Um guia completo e prático de agilidade

SCRUM 360

JORGE AUDY

Jorge Audy

Escoteiro e agilista. Mestre em administração na linha de pesquisa de Gestão da Informação, formado em 1986, concursado nos anos 80, empresário nos 90, coordenador de desenvolvimento na ADP Brasil nos anos 2000 e no Grupo RBS em equipes corporativas e de produtos digitais até 2013, desde então consultor na DBServer. Praticante de métodos ágeis desde 2008 e como Agile Coach desde 2011, palestrante e participante ativo em eventos sobre métodos ágeis, equipes de alta performance e, mais especificamente, SCRUM.

Prefácio

Desde 2011, venho compartilhando na web princípios e métodos ágeis. Neste guia apresentarei uma visão prática sobre Agile e Scrum, a partir da seleção dentre mais de 300 posts publicados sobre agilidade e cotidiano em http://jorgekotickaudy.wordpress.com e em minha coluna no site de conteúdo http://www.baguete.com.br.

A estrutura e organização deste guia reforçam a ideia de que 75% do entendimento e sucesso na adoção de qualquer método recaem sobre pessoas e autoconhecimento, o quanto elas entenderam e praticam de fato ou quanto é meramente operacional ou marqueteira. Ao iniciar um projeto, workshop, evento ou mais um dia de trabalho, é fundamental que saibamos por que estamos ali, qual a expectativa de valor.

Este livro compartilha 25 anos de mercado, quatro deles em metodologias ágeis. O objetivo deste livro é mostrar a prática e os princípios por trás de um time Scrum, conceitos vitais para o entendimento e sucesso de todo e qualquer time ágil. Para fazer certo tem que entender os porquês. A cobertura é de 360°, teoria, princípios, métodos, técnicas, pessoas, antes e depois, com lições aprendidas, sem meias palavras, característica recorrente no blog e no baguete. A teoria todos sabem, o difícil é o que não está escrito: aqui mostro tudo aquilo que não está no Scrum Guide.

Apresentação

Nos últimos anos, o desenvolvimento de software tem se tornado um componente vital nos negócios das empresas. Profissionais, empresas (públicas e privadas) e negócios dependem, de forma crescente, de software para decisões estratégicas, táticas e operacionais. Por este motivo, devido ao papel cada vez mais estratégico do software para as empresas, o desenvolvimento de software precisa ser executado em um ritmo cada vez mais acelerado e está sujeito a uma cadeia de mudanças que são inerentes ao processo.

As condições de mercado se alteram rapidamente, as necessidades dos usuários finais variam e fatos novos de mercado emergem sem aviso. É preciso ser rápido o suficiente para dar uma resposta adequada ao ambiente de negócio, o que exige maior fluidez, flexibilidade e capacidade de adaptação a mudanças. Neste ambiente, a baixa qualidade tem se tornado cada vez menos tolerável.

O software por sua própria natureza deveria ser projetado para se adaptar a essas mudanças. Porém, os desafios para o desenvolvimento de software vão muito além dos aspectos técnicos. Fatores organizacionais, culturais, pessoais, econômicos e legais são exemplos de aspectos não técnicos que contribuem para tornar o desenvolvimento de software uma atividade relativamente complexa. Desenvolver software é difícil, e desenvolver software de qualidade e com valor agregado para o cliente é ainda mais desafiador.

Temos observado nos últimos anos um crescente interesse pelas metodologias ágeis de desenvolvimento de software. O principal objetivo dessas metodologias é compartilhar práticas de desenvolvimento de software de forma alinhada aos desafios da sociedade moderna. Seu surgimento, no final da década de 90, formalizou-se com o manifesto ágil em 2001.

O Scrum é uma das metodologias ágeis mais conhecidas e mais utilizadas na indústria atualmente. Seu foco é na gestão de ciclos iterativos de desenvolvimento de projetos e tem como premissa a inspeção e a adaptação constantes. É relativamente simples e fácil de aprender, porém, difícil de executar, pois não basta apenas aprender a técnica. É preciso entender e enxergar desenvolvimento de software através de uma perspectiva diferente, alinhada com tudo o que escrevi nos parágrafos iniciais desta apresentação. E é neste contexto que surge esta obra escrita pelo Jorge Horácio Audy.

Por alguns anos tive a oportunidade e o prazer de ministrar conteúdo relacionado a gerenciamento de projeto de software no Bacharelado em Sistemas de Informação da Faculdade de Informática da PUCRS. Scrum era um dos tópicos abordados em aula. Sempre que possível, eu dizia aos alunos que as técnicas, conceitos e ferramentas que eles estavam aprendendo formavam o conjunto mínimo necessário de informações que eles deviam conhecer. O que faria mesmo a diferença era a forma como eles iriam atuar no dia a dia, considerando as pessoas e os ambientes onde elas estão inseridas, os imprevistos, os problemas e os desafios comuns que são inerentes ao desenvolvimento de software nos dias atuais. Entretanto, raramente vemos isto documentado em livros, sejam eles escritos para apoio ao ensino ou voltados para os profissionais de determinada área.

O que o Jorge faz neste livro é exatamente o que não vejo em outras obras sobre Scrum. Com um embasamento teórico refinado e com diversos exemplos práticos, o autor nos brinda com um texto fácil e prático, recheado de dicas, conselhos e recomendações, tendo como base sua experiência recente e intensa de trabalho com Scrum e metodologias ágeis como um todo. Em resumo, é literalmente uma visão em 360 graus do Scrum, onde o autor compartilha toda sua vivência com a prática da metodologia, buscando promover e fomentar o aprendizado vicário na sua forma mais pura e direta. Afinal, conhecer técnicas, ferramentas e conceitos é o mínimo. O que faz mesmo a diferença é nossa capacidade de interpretar, adaptar para a nossa realidade e executar aquilo que conhecemos e aprendemos. Boa leitura!

RAFAEL PRIKLADNICKI

Professor da Faculdade de Informática da PUCRS, Diretor do Parque Científico e Tecnológico da PUCRS (TECNOPUC), Coordenador do Grupo de Usuá-

rios de Metodologias Ágeis do RS (GUMA-RS)

Sumário

Capítulo 1

Modelo mental

A vida é feita de mudanças e ritos de passagem, repleta de simbolismos e significados. Podemos ter sorte, mas quanto maior nosso nível de consciência e preparo para enfrentar cada nova etapa, maior a chance de sucesso. Vamos começar refletindo um pouco sobre isto, quem somos, o que e por que fazemos, como percebemos e somos percebidos, o quanto é consciente ou inconsciente.

1.1 Escotismo

"Se tiver o hábito de fazer as coisas com alegria, raramente encontrará situações difíceis"
– Sir Robert Baden-Powell

Uma obra que compartilha muito mais que princípios e métodos ágeis, mas a convicção de que é possível termos melhores resultados por meio de

um ambiente de trabalho mais saudável e inteligente.

Eu não apreendi agilidade a partir de um curso ou experiência, foi a partir da percepção de que eu seria mais feliz se conseguisse agir no trabalho da mesma forma que agia aos finais de semana no escotismo.

O papel de um chefe escoteiro não é o de chefiar, ele lá está para orientar. Não para impor, mas para facilitar o aprendizado ouvindo a opinião dos jovens e permitindo que eles opinem, discutam, escolham e experimentem, tentem de diferentes formas e aprendam com isto. Aprender fazendo!

Uma seção escoteira possui planejamento de ciclos, divididos em meses, com um planejamento específico para cada sábado. Cada ramo possui suas reuniões de autogestão, os lobinhos em Rocas de Conselho, escoteiros e seniores em Cortes de Honra, os pioneiros em Assembleias, todos exercendo uma liberdade com responsabilidade.

Cada jovem possui habilidades e agrega mais em alguma função, todos possuem direitos e deveres frente à sua equipe. Acima de tudo, não se acomodar, buscar superar seus limites, com segurança, um apoiando o outro, comemorando os acertos e aprendendo com os erros.

Se cada um trabalhar da mesma forma como um jovem pratica escotismo, a agilidade será mera consequência.

1.2 Muros e feudos

"Os membros de uma equipe vencedora lutam contra os seus concorrentes. Os membros de uma equipe perdedora lutam entre si."
– Joseph M. Juran

Já está na hora de todas as empresas derrubarem os muros e defesas existentes entre as diferentes áreas e equipes envolvidas na indústria de software. O que ainda vemos são fortificações em que as áreas competem entre si.

Não existe "minha parte" e "parte deles", existe um serviço a ser realizado ao cliente. A solução não é difícil: é preciso estabelecer um processo em que cada equipe enxergue seu papel no todo, com muita interação, iterações e antecipação.

A proposta é termos uma única perspectiva de negócio, todos cientes e envolvidos. Ainda hoje o que se vê é o estabelecimento de feudos, cada qual

preocupado a não ser o culpado, fazendo a sua parte. É preciso mais que isso, é preciso ter uma visão holística, sinérgica, colaborativa.

Fig. 1.1: Muros e feudos

Devemos lançar mão dos meios e estratégias possíveis e disponíveis para estabelecer a melhor comunicação e convergência. Uma empresa é como um relógio: de nada adianta uma peça ser de ouro se o encaixe nas demais peças não for preciso; todas têm que trabalhar em sincronia e em função de um único objetivo.

Acredito na desconstrução de muros e conveniências, em ecossistemas colaborativos, onde todos façam o seu melhor, para ter orgulho do resultado final. Métodos ágeis não são mágicos, dependem de interesse, engajamento, trabalho duro, antecipação e comunicação. A solução real passa por mobilização, capacitação, coaching, realismo e bom senso.

1.3 Desarmando rótulos e paixões

"Falta de tempo é desculpa daqueles que perdem tempo por falta de métodos."
– Albert Einstein

Gosto de dedicar tempo para analisar os comentários que recebo em artigos ou escuto em eventos, tentando entender o que tem por trás de cada apego, resistências e muros.

Tenho convicção de que o problema são as paixões humanas na busca da polarização por rótulos. Tendemos a ser 100% a favor ou contra algo, capitalismo, socialismo, anarquia, agilidade, construtivismo, como se tudo não tivesse coisas boas e ruins. Ao escutarmos qualquer rótulo que represente mudanças, entramos em modo defensivo.

Muitas pessoas assistem a palestras ou cursos sobre métodos ágeis e não percebem os alertas de que não é bala de prata, que não é a derradeira solução para todos os problemas. Métodos ágeis oportunizam que os problemas sejam percebidos antecipadamente, destacam falhas na comunicação e geram a oportunidade para que todos os envolvidos contribuam de fato na estratégia, o restante é trabalho duro a cada dia.

Não tenho certeza se é do ser humano, latino, brasileiro ou gaúcho, mas adoramos polarizar, somos a favor ou contra o rótulo. Não importa se o rótulo é tangível ou intangível, se é uma corrente de pensamento, método, pensamento, time de futebol, partido político etc., chega a ser irracional.

Muitas pessoas NÃO me questionam quanto a uma das técnicas, um conceito, uma prática, aplicabilidade de um tipo de reunião ou artefato, quais são os pontos prós e contras de uma dinâmica. Apenas questionam o rótulo e normalmente são contra tudo, como se fosse tudo ou nada.

1.4 Sociedade industrial ou do conhecimento

"Toda empresa precisa de gente que erra, que não tem medo de errar e que aprenda com o erro."
– Bill Gates

Estamos tentando migrar de uma sociedade industrial, baseada na Administração clássica e mecanicista de Taylor e Fayol, para uma sociedade do conhecimento. As principais escolas apontam a necessidade de priorização das pessoas, instigando o senso de pertença e motivação para fazerem mais e melhor, atraindo e retendo talentos criativos e empreendedores.

Desde os primórdios da revolução industrial, é comum trabalhar doze horas em um dia, sempre com prioridade na quantidade e não na qualidade, desconsiderando os efeitos da fadiga. No fim, a pessoa que exige carga ho-

rária dobrada e pressão desmedida é a mesma que assinará os cheques para concertar, manter o legado e refazer mais adiante.

Esta realidade industrial treinou e ainda treina gerações de profissionais frustrados, que se contentam em fazer muitas coisas ao mesmo tempo com baixa qualidade. Mudar este modelo mental exige esforço e necessita que prestemos atenção a cada oportunidade de reiterar o valor dessa mudança.

O segredo está no senso de pertença. Se antes éramos pagos para executar e não questionar, um novo modelo mental espera que as pessoas se interessem, que participem, dediquem-se à construção de algo que dê orgulho a todos os envolvidos, gerando valor para a empresa e para si mesmo.

1.5 Autoconhecimento

"Os problemas significativos que enfrentamos não podem ser resolvidos no mesmo modelo mental que os criou."
– Albert Einstein

Métodos ágeis, *management 3.0*, equipes de alto desempenho, está na hora de tentar fazer do trabalho um local que mereça nossa presença por mais de 10 horas a cada dia, considerando o deslocamento e almoço. O primeiro passo é o autoconhecimento. Entender como você e outras pessoas se comportam no trabalho. Exemplificarei sob a ótica da felicidade:

- **Apaixonados**: estão onde querem estar, creem, querem crescer e colaborar. São minoria; fazem seu melhor e sentem-se felizes mesmo sem reconhecimento.
- **Bem-resolvidos**: racionais e focados, agregam valor e fazem a diferença. Enquanto estiverem ali, farão seu melhor, agregando a eles e à empresa.
- **Acomodados**: não refletem muito sobre o assunto, está bom como está. Estão satisfeitos fazendo a sua parte, conforme lhes foi proposto ou ordenado.

- **Indecisos**: projetam sua ansiedade e inquietação na empresa e nos colegas. Ansiosos com a função, cargo, processo, alternam sua posição e opinião.
- **Insatisfeitos**: orgulham-se de suas insatisfações, mas sem abertura a debates. Não procuram soluções e mesmo assim vão desconstruindo as coisas ao redor.
- **Problemáticos**: são aqueles com temperamento que frequentemente geram problemas. Inconscientemente vão provocando ruído e desconforto sem objetivo.

Navegamos entre diferentes perfis, mas se eu pudesse optar, elegeria um time inteiro de bem-resolvidos. Devemos dar nosso melhor no lugar onde estamos, de forma racional, madura, isenta e objetiva. Se não temos orgulho daquilo que, e com quem, fazemos, façamos mesmo assim o nosso melhor. Desta forma, vamos aprender e crescer. Mesmo em um ambiente difícil, isso nos dará maturidade e visibilidade que abrirão oportunidades para chegar aonde queremos, o que às vezes é outro lugar.

1.6 Não mudo nada porque não posso mudar tudo

"A alegria que se tem em pensar e aprender faz-nos pensar e aprender ainda mais."
– Aristóteles

Graduei-me em 1987. Desde então vejo empresas investirem em certificações como 5S, 6Sigma, CMMI, ITIL, MSF, PMBOK, MPS-BR. Agora é Lean, Scrum, XP, Safe, todas prometendo ser a derradeira solução e talvez todas sejam!

Mas não existe receita prescritiva garantida. A solução é adaptativa. Dentre as alternativas conhecidas, opte por aquelas que mais casam com os seus valores. Esforce-se em adotá-las e adaptá-las à sua realidade, contexto, história e pretensões, valorizando a cultura corporativa e as pessoas.

Quanto à adoção de métodos ágeis, o fato de uma empresa não conseguir implantar 100% de agilidade não a impede de melhorar. Quem se diz ágil, mas só aceita mudar tudo ou nada, precisa enxergar a mudança com agilidade, com releases, iterações, cadência e melhoria contínua.

A minha sugestão é de que agilidade não é só para gerenciamento de projetos: é para tudo, é para a vida. Veja a empresa, setor, equipe e carreira como algo que merece uma abordagem ágil. Dê o exemplo, seja solução, não problema.

Quer validar essa percepção? Participe de eventos, palestras, grupos de usuários e comunidades de prática, e perceberá que não está sozinho. Estamos todos no mesmo barco, estamos tentando, evoluindo, aplicando agilidade na agilidade.

1.7 Felicidade

"O segredo do sucesso não é prever o futuro. É preparar-se para um futuro que não pode ser previsto."
– Michel Hammer

Muitos se enganam quanto à busca da felicidade no trabalho. Alguns confundem vídeos e livros mostrando o lado legal de trabalhar em empresas ágeis, relatos e posts de agilistas surfando, andando de bicicleta, saindo com os amigos, fazendo esporte, e deduzem que é só isso.

Queremos ser felizes, mas a felicidade no trabalho está atrelada a resultados, mesmo que muitas pessoas vejam empresas descoladas e se iludam achando que eles só se divertem e não prestam contas por resultados. Na minha visão, temos nosso tempo dividido em três partes iguais:

- **1/3 Descanso (sono)** – Estou partindo do princípio tradicional de 8 (oito) horas de sono diário, ciente de que flutua com a idade e opções pessoais;
- **1/3 Lazer** – Quando não estamos dormindo ou trabalhando, temos total liberdade para aproveitar (ou desperdiçar) ao máximo nossos dias;

- **1/3 Trabalho** – Temos que buscar um ambiente que nos proporcione ou pelo menos não impeça nosso crescimento e satisfação, com foco em carreira.

O maior objetivo no trabalho é construir uma carreira, o que difere do lazer, pois trata-se de um contexto em que existem direitos e deveres com diferentes interesses e interessados. Só dá certo se entendermos a felicidade como crescimento e como parte de algo maior, antecipando, aprendendo e fazendo a diferença.

Ser feliz no trabalho é colaborar e contribuir na construção de um time vencedor, ver sua carreira decolar inserida na organização. Mas, eventualmente, também é saber a hora de pular de um barco que não o valoriza e só busca resultado a qualquer custo.

1.8 Individualismo ou coletivo

"Inteligência é a capacidade de se adaptar à mudança."
– Stephen Hawking

Em uma equipe ágil não há espaço para estrelismo. Se você quer ser o centro das atenções ou fica reiteradamente colocando alguém da equipe em foco, pró ou contra, reveja seus conceitos. O objetivo é o equilíbrio do coletivo, na certeza de que o crescimento, ideias e boas iniciativas estão abertos a todos para participarem e colaborarem com a melhoria contínua.

Acredito em abordagens construtivistas, especialmente no valor de se trabalhar verdadeiramente em equipe, com claro entendimento do que isto significa. Devemos ter orgulho do que e como estamos trabalhando e construindo. Afinal, passamos mais tempo com nossos colegas de trabalho do que com nossas famílias, isso diz tudo!

O primeiro passo é conhecermos melhor aqueles com quem trabalhamos cotidianamente, quer sejam nossos colegas de equipe, empresa, clientes e fornecedores. Qual o maior ganho na realização de retrospectivas, reviews, debates e outras dinâmicas ágeis? É para nos aproximarmos uns dos outros, deixando de ser apenas mais um.

Cabe a cada um de nós o papel de conhecer e valorizar a contribuição de cada um, investir no potencial de crescimento de todos, equilibrar as diferenças que sempre existirão em um time, para assim torná-lo mais forte e vencedor. Não se tem sinergia em meio a conflitos e desconfianças, apenas na colaboração e convergência.

O pior caminho é ficar criticando ou elogiando alguém em demasia. Não há ser humano que aguente de forma isenta um bombardeio de responsabilização, prós ou contras. Após inúmeros "Você não tem jeito" ou "Você é o melhor", o tal cidadão corre o risco de começar a acreditar nessas bobagens e acabará metendo os pés pelas mãos.

O ambiente de trabalho refletirá os valores que cultivamos na nossa vida particular, mas esta é uma via de mão dupla. Podemos melhorar em ambos se começarmos a repensar nossos valores de forma positiva. Pensar em todos os envolvidos, em si como parte e não como todo, evitando o embate inócuo e desnecessário.

Não é fácil para ninguém. Devemos contar uns com os outros, pois se o time vencer, todos vencem. Se o ambiente é saudável, se de fato somos ágeis, não precisamos nos preocupar em sermos "vistos" ou não. A construção é de todos, mas a contribuição pessoal acabará nos distinguindo.

1.9 Confiança e melhoria são como uma poupança

"Nós somos aquilo que fazemos repetidamente. Excelência, então, não é um modo de agir, mas um hábito."
– Aristóteles

A TI escolheu mudar a metodologia e tem a convicção de que isso vai ser bom para todo mundo; decisão esta, que aconteceu após décadas trabalhando em modelos mais tradicionais e comando-controle. É preciso algum tempo para aprender a trabalhar diferente e mostrar a que viemos. A prioridade deve ser aproveitar os aprendizados iniciais para que resultados comecem a ser percebidos e comecemos a formar um novo modelo mental.

Acima de tudo, temos que nos conscientizar de que a confiança da empresa e clientes é como uma poupança: você começa a depositar pequenos

valores, que vão sendo corrigidos e, com o tempo, vão se avolumando e depois poderemos sacar. É um ciclo virtuoso. Algumas vezes o cliente compra a ideia desde o inicio, mas normalmente eles precisam ver acontecer para acreditar.

Mais que isso, um banco em que você possui uma boa poupança lhe facilitará o crédito. Estabelece-se um modelo de reciprocidade, dada a confiança e boa relação de negócios. Com o cliente é igual, ele lhe dará tanto crédito quanto você estabelecer uma boa relação e gerar bons negócios para ele. Se ele começar a receber software de valor a cada duas semanas, começará a confiar cada vez mais em suas orientações.

Não espere sanar décadas de maus hábitos e débitos técnicos em algumas semanas. Ouço desenvolvedores exaltados dizendo que estão fartos de não terem a oportunidade de fazer melhor. Esquecem que algumas semanas antes havíamos iniciado um processo de mudança e que ela envolve questões culturais e de confiança que precisam ser estabelecidas em outros patamares. Isso leva tempo e exaltar-se e polarizar não ajuda em nada, pelo contrário, só posterga seus objetivos.

Pedir e ficar reclamando é um resíduo histórico dos tempos de comando-controle, quando nos restava apenas reclamar. Agora a responsabilidade é nossa em gerar argumentos, negociar, gerar fatos, agilidade na agilidade, pois a melhoria é iterativo-incremental também para o processo.

1.10 Mudança de hábito

"O hábito nunca é bom, nem sequer o hábito de fazer boas ações. As boas ações, quando se tornam hábito, deixam de ser atos de virtude. O verdadeiro bem só é alcançado com esforço."
– Immanuel Kant

Nosso cérebro não é capaz de lidar com o imenso volume de informação que nossos cinco sentidos captam a cada milésimo de segundo. Para resolver esta restrição, ele tenta trabalhar somente com a mudança. Para isto ele abstrai e deduz muito do que acreditamos estar sentindo.

Somos a compilação de nossos hábitos desde a hora que acordamos, tomamos café, nos deslocamos para trabalhar, trabalhamos e assim por diante.

Quantas vezes nos perguntam sobre algo que esteve à nossa frente e que não percebemos?

Aprender algo novo exige desaprender o antigo, isso consome energia e recursos, um processo que pode gerar ansiedade e ativar nossas defesas psíquicas, dificultando nossa percepção e raciocínio crítico para realizar a mudança com sucesso.

No livro *O poder do hábito*, de Charles Duhigg, aprendemos que cada hábito possui um **gatilho** que nos faz entrar no piloto automático, e executar uma **rotina** que fazemos sem refletir. Finalmente, há a **recompensa**, que mostra que esse hábito vale a pena ser utilizado novamente no futuro.

Refletir sobre hábitos me faz lembrar artigos sobre PNL (Programação neurolinguística), mudança, parceiros de viagens, feedback e retrospectiva, sobre o entendimento das dezenas de teorias que nos ajudam a entender os mecanismos psicológicos e sociológicos em que estamos imersos.

Os piores hábitos são aqueles que não trazem uma recompensa real, ou seja, quando o resultado atingido é uma falsa impressão criada pelo nosso inconsciente. A mudança de maus hábitos, como inércia, sedentarismo, procrastinação, fumo ou álcool depende da substituição da rotina que nos leva a essas pseudo-recompensas.

1.11 Morte às baias e aos gaveteiros

"As oportunidades para melhorias existem em grande quantidade, mas não mandam aviso."
– Joseph M. Juran

Morte às baias e aos gaveteiros, verdadeiras trincheiras, feudos onde seus senhores se protegem. Chega da falsa sensação de segurança atrás de painéis estofados de cor bege, mesas oblíquas que na década de 90 foram adquiridas por verdadeiras fortunas, cheias de regulagens e configurações individualistas.

Morte às dezenas de salas em um único andar, paredes desnecessárias, armários e quilos de papéis guardados por anos e já amarelados, que nunca foram acessados neste ínterim, fim às tralhas armazenadas, sobre as quais, quando perguntamos, ninguém sabe de quem são e para que servem.

Cada vez mais empresas investem em ambientes abertos, com grandes mesas onde gestores e equipes sentam-se lado a lado. Em contraposição, alguns se agarram a mesas individuais a uma distância segura de dois metros das mesas mais próximas, sonhando um dia ter uma sala envidraçada, uma sala de chefe.

A cada ano se consolidam ambientes mais amplos, abertos, pouco espaço pessoal e muito espaço comum, mesões coletivos sem gaveteiros nem divisões, cada vez mais laptops em vez de desktops, ambientes descontraídos, e cada vez mais inspiradores com seus espaços de descompressão e cozinhas.

Se por um lado aproxima, simplifica e otimiza a comunicação e interação humana, é um grande desafio para o bom senso de cada um. Ambientes ágeis do século XXI exigem mais disciplina que salas, baias e mesas individuais do século XX, afinal, há direitos e deveres.

Cada um possui espaço em um locker, para poucos itens pessoais protegidos. O restante é alçada dos times definir as regras de convivência, desde o ruído ao uso de espaços e objetos comuns, um treino diário de auto-organização.

Ambientes ágeis e inspiradores são como os novos condomínios nos quais moramos: os apartamentos são menores do que sonhávamos no passado, mas o espaço comum tem espaço kids e gourmet, sauna, piscina, fitness, cyber café etc.

Empresas que são os ícones desta nova ordem, como Google, Facebook, Yahoo, Pixar, Apple etc. têm muito mais que isso em todos os aspectos. Elas vão além, com espaços que são verdadeiros playgrounds, mas que vêm acompanhados de outras responsabilidades.

O importante é que esse caminho não tem volta, assim como o uso permitido de bermudões e sandálias, cancelando exigências insanas de roupa social em épocas do ano com temperaturas de até 40° em um país tropical.

1.12 Equilíbrio

"Viver é como andar de bicicleta: é preciso estar em constante movimento para manter o equilíbrio."
– Albert Einstein

O segredo da agilidade é o equilíbrio, como no Yin-yang do Taoísmo, forças fundamentais, opostas e complementares. Olhar só por uma lente sempre nos leva à miopia, afinal polarizações e simplificações são ingênuas:

Individualismo x coletivo

Tudo na agilidade necessita do coletivo. É a forma de potencializar cada momento e resultados, a partir daquilo que de melhor cada um tem para contribuir. Mas o coletivo não pode abafar o pensamento e opinião individual; o contraditório é a receita de um time vencedor.

Qualidade x entrega

Há uma eterna e saudável discussão entre cumprir o prazo e subir a régua da qualidade. O embate entre resultado hoje e continuidade amanhã, qualidade e orgulho, não é para ser dolorido; deve ser provocativo, bem argumentado. É buscar o equilíbrio de ontem, de hoje e de amanhã.

Conforto x mudança

As pessoas têm medo de mudar, mesmo para melhor, arraigadas às suas zonas de conforto. Porém, cada vez mais pessoas e empresas começam a repensar seus valores, lutando contra a mesmice, o fazer mais do mesmo como sempre foi feito, cientes de que poderiam tentar diferente.

Felicidade x resultado

Os conceitos de ambiente inspirador, felicidade, descompressão e orgulho estão atrelados a resultados. Nada disso tem valor se não os usarmos para superação, inovação, empreendedorismo e entrega contínua de valor. Felicidade não quer dizer lazer, o objetivo ainda é o negócio.

Inovação x domínio

A inovação não vem da pressão, mas da inspiração de diferentes fontes; é sair da caixa, é o ócio criativo. A inovação pressupõe instigar a mente e aceitar o risco de tentar e errar provavelmente mais errar do que acertar mas, por outro lado, tem horas em que não dá para arriscar.

Transparência x educação

Esse é um dos embates mais importantes. Já vi de tudo: prepotência, destratar estagiário, sair e bater a porta no meio de uma discussão, ataque pessoal no meio de um debate de ideias. Para estes, sugiro rever seus conceitos e ler um bom livro de etiqueta no trabalho. Transparência tem muito a ver com bom senso e educação.

Planejado x extras

A aceitação de mudanças, mesmo no final do Sprint é uma provocação, pois a aceitação depende de racionalidade, de real valor para o negócio e viabilidade. A equipe sempre deve ter o senso de pertencimento e questionar, pois se cada um sente o projeto como seu, vamos querer fazer o que é certo.

Multidisciplinaridade x especialista

Cuidado, precisamos de especialistas. A construção de produtos diferenciados exige profundidade de conhecimento. A multidisciplinaridade desejada é para desimpedir gargalos. Não queremos ou esperamos que todos façam tudo sempre, mas queremos especialistas com flexibilidade.

Capítulo 2

Sobre os ombros de gigantes

Na vida e no trabalho, fui acumulando interesses originados na psicologia, ciências sociais e outras fontes teóricas para o entendimento dos porquês. Leia com calma. O entendimento destes fundamentos lhe dará excelentes argumentos para o seu cotidiano, pois, como dizia minha orientadora no mestrado "veremos mais longe se estivermos sobre os ombros de gigantes", grandes pensadores, pesquisadores, estudiosos. No campo da gestão da informação, por exemplo, há um site com as principais teorias utilizadas http://istheory.byu.edu

2.1 Lei Yerkes-Dodson (1805)

"Apoie-se em uma gestão visual com total transparência de procedimentos, processos e valores."
– Kaizen, Masaaki Imai

Uma pesquisa realizada em 1908 pelos psicólogos Robert Mearnes Yerkes e John Dodson Dillinghamm, que acabou sendo conhecida como Lei de Yerkes-Dodson, demonstrava que nossa performance é afetada positivamente por eventual estado de excitação fisiológica ou mental.

Yerkes e Dodson apresentaram um gráfico com a influência positiva causada por níveis de excitação até um teto, a partir do qual o desempenho passa a diminuir. O processo é ilustrado como uma curva em forma de U invertido.

Além dessa conclusão, também perceberam e registraram que há uma sensível diferença nesta curva quando o desempenho medido diz respeito a tarefas ou desafios mais ou menos complexos. As tarefas menos complexas possuem um teto mais alto e mais sustentável, enquanto nas mais complexas é mais baixo.

Em TI, onde lemos "excitação" devemos interpretar como pressão por prazos exíguos, orçamento apertado, grandes desafios etc. Aqui chega o ponto onde sou obrigado a perguntar: em qual das curvas será que boa parte das tarefas de desenvolvimento de software se enquadra?

- Tarefas simples – exigem atenção, acesso à memória rápida, aplicação de boas práticas, um mínimo de previsibilidade e risco moderado.
- Tarefas complexas – exigem atenção dividida, memória de trabalho, encadeamento de tomadas de decisão, multitarefa e adaptabilidade.

A lei de Yerkes-Dodson tem tudo a ver com ambientes profissionais exigentes e tensionados. É uma teoria da psicologia que nos ajuda a explicar por que o excesso de pressão e stress acabam gerando bugs, dívida técnica crescente, falta de qualidade e sistemas que passam a ser encarados como legados logo após entregues.

Alguns estudos, utilizando esta teoria, avaliam o quanto a pressão de orçamentos e cronogramas afetam, positiva ou negativamente, a produtividade. Em projetos de software, manter um nível benéfico de excitação requer a cooperação entre clientes e equipes, onde a colaboração e a auto-organização geram melhores resultados que pressão vertical.

2.2 Ciclo de Deming (1950)

"A gente não se liberta de um hábito atirando-o pela janela: é preciso fazê-lo descer a escada, degrau por degrau."
– Mark Twain

Se você tem restrições quanto ao Scrum e outros métodos ágeis, quem sabe isso não muda se adotar o velho e conhecido PDCA (*Plan-Do-Check-Act*), que é consensual e também está relacionado ao Japão pós-guerra? O ciclo mais conhecido de melhoria contínua e incremental foi idealizado por um americano, mas bancado pelo governo japonês no esforço de reconstrução do pós-guerra.

William Edwards Deming é um estatístico americano que revolucionou o *mindset* japonês e tornou-se o precursor e nome mundial dos programas de qualidade total, seguindo os passos de Walter A. Shewhart no uso de estudos estatísticos de processos. Ele desenvolveu 14 pontos para a gestão da qualidade total, a qual deve ser continuamente aperfeiçoada:

1) Trabalhar para o aperfeiçoamento de produtos e serviços, perpetuando-os, mantendo e gerando empregos;

2) Adaptar-se à nova era econômica;

3) Trocar a necessidade de inspeção pela implantação de uma cultura de qualidade total, envolvendo todos em todo o processo;

4) Racionalizar a cadeia de produção, inclusive fornecedores e parceiros, criando relacionamentos duradouros, baseados na qualidade e na confiança;

5) Adotar melhoria contínua em todo o processo, maximizando qualidade e produtividade, reduzindo ao máximo os custos;

6) Fomentar o próprio desenvolvimento com treinamentos frequentes no seu local de trabalho, com foco no desenvolvimento das pessoas e do processo;

7) Adotar o modelo de Coaching, ajudando as pessoas a realizar um trabalho melhor (é preciso para isto evoluir todo o modelo mental de liderança);

8) Eliminar o medo;

9) Investir na construção de um ecossistema, quebrando os silos e a competição entre diferentes interessados;

10) Eliminar slogans e exortações aos empregados;

11) Eliminar métricas artificiais para o operariado, evitar administração por objetivos (APO) e administração através de números e metas numéricas;

12) Deve haver senso de pertencimento e poder nas pessoas em relação ao seu trabalho e ao resultado deste;

13) Valorizar todos os perfis e profissionais envolvidos direta ou indiretamente no produto ou serviço prestado; o orgulho deve ser de todos;

14) As mudanças, melhorias e transformações são uma tarefa e meta de todos.

2.3 Pareto e Juran (1951)

"A qualidade é avaliada pelo usuário ou cliente. O objetivo é satisfazer o cliente com a quantidade certa. Nem mais nem menos."
– Joseph M. Juran

O economista italiano Vilfredo Pareto desenvolveu em 1906 um modelo que buscava descrever a desigualdade na distribuição da riqueza na Itália, concluindo que 20% das pessoas detinham 80% da riqueza. Conhecido como Princípio de Pareto, foi Joseph Juran que levou este estudo à esfera organizacional, segundo o qual 80% dos problemas são causados por 20% das causas, enfatizando a relevância de focar em valor.

A regra 80/20 do estudo de Pareto sobre a concentração da riqueza italiana para no máximo 20% da população foi migrada por Juran para o mundo corporativo. Juran sugere focarmos nos 20% que resolvem 80% do objetivo. Devemos trabalhar naquilo que gera resultados e é relevante, em vez de trabalhar ou investir tempo e energia em coisas irrelevantes ou de pouco retorno.

Um indicativo relevante, quer seja de um país, empresa ou pessoa, é quando estamos sempre atrás do prejuízo, apagando incêndios, fazendo mais

do mesmo. Isto pode indicar que o foco está nos 80% triviais e não nos 20% vitais. Bombeiros corporativos são valorizados, resolvendo problemas imediatos, mas raramente alguém questiona por que isso foi necessário. Às vezes o próprio bombeiro foi o gerador do incêndio.

O mais importante não é fazer a ponte, mas fazer a ponte no lugar e na direção certa. O que queremos é fazer a coisa certa na hora certa, buscando entender e atender os 20% mais vitais que geram 80% dos resultados desejados.

JMS *Juran Management System* é um modelo de qualidade que colaborou para a configuração da cultura Lean. Ele propôs três pilares da qualidade, com foco em redução de desperdícios e geração de valor, princípios Kayzen de melhoria contínua e equipes auto-organizadas:

- Planejamento da qualidade com entendimento real da necessidade do cliente em ciclos iterativo-incrementais;
- Controle de qualidade em cada passo, durante as operações, com mínima inspeção;
- Melhoria da qualidade através de aprendizado contínuo e incorporado ao processo (*breakthrough*).

Sobretudo, Juran defendia a importância do fator humano e uma mudança cultural dos gestores, que devem integrar-se mais à estratégia de sua força de trabalho, reiterando a iniciativa de garantir os três pilares da qualidade.

2.4 Teoria da dissonância cognitiva (1957)

"Aquele que não tem confiança nos outros, não lhes pode ganhar a confiança."
– Lao-Tsé

A Teoria da Dissonância Cognitiva, cunhada por Leon Festinger e Carl Smith, afirma que o nosso consciente, ao não encontrar explicação ou solução para uma angústia, gera uma dissonância, e mecanismos psíquicos de defesa do nosso inconsciente se encarregam de resolvê-la.

A psicanálise apresenta uma lista de defesas psíquicas que corroboram essa teoria, quando o inconsciente justifica o consciente através de defesas psíquicas. Devemos ter cuidado com o abuso, por exemplo, justificar tudo como culpa dos outros. A busca por culpados é uma fuga da solução.

- **Negação**: é negar a existência de um problema nos convencendo que ele não nos afeta, eliminando assim eventual angústia ou sentimento de culpa.
- **Projeção**: é quando culpamos alguém por algo que poderíamos ter evitado ou resolvido, uma forma de nos defendermos desta angústia;
- **Transferência**: é projetar emoções em alguém porque não conseguimos resolvê-las de outra maneira;
- **Racionalização**: é quando buscamos uma explicação aceitável para uma atitude indesejada, justificando ou contornando para nos isentarmos;
- **Substituição**: é quando substituímos um sentimento por outro mais conveniente, que alivie uma angústia ou ansiedade;
- **Identificação**: é buscar aproximação com seus opostos para não precisar opor-se a eles, é encontrar um ponto em comum que nos identifique;
- **Repressão**: evitar algo que já vivenciamos e não gostamos, é optar pela omissão para evitar reviver uma situação indesejada;

2.5 Teoria da contingência (1958)

"À medida que a tecnologia avança, as empresas utilizam inicialmente uma estrutura mais mecanicista e depois uma estrutura mais orgânica."
– Joan Woodward

A essência da Teoria da contingência ou Teoria contingencial é não haver um único modelo organizacional ideal. Qualquer modelo está sujeito a

se contingenciar frente à sua realidade interna e externa. Uma experiência de sucesso deve ser estudada, mas é temerário apenas copiá-la sem entender o contexto onde foi aplicada *versus* as características da sua própria organização.

Os gestores da Toyota afirmavam que, quanto mais os americanos os visitavam e tentavam copiar seus processos, mais se distanciavam das razões de seu sucesso. O segredo estava na interação entre a cultura organizacional e a microcultura de seus times.

A Teoria contingencial não afirma que o processo, ferramental e estruturas sejam irrelevantes, que o estilo de liderança não é fator crítico de sucesso, mas garante que os motivos do sucesso organizacional são resultado da equação de fatores e atores, internos e externos, cultura e ecossistema. Essa ideia originou-se do contraponto a estudos do início do século XX que propunham uma forma ideal de organização, da gestão a produção.

Mas cada empresa reage diferentemente a condições internas e externas, gerando oportunidades e riscos, facilidades ou restrições.

Ambiente pode ser entendido como o contexto onde a organização está inserida e que constantemente a influencia, como condições ecológicas, culturais, políticas, econômicas, legais e tecnológicas. Cada uma destas variáveis e outras mais interferem nas organizações, interagindo com elas mesmas, potencializando-se, outras se anulando, cabendo a nós uma análise contínua, de forma a agilizar a adaptação, quer para proteção ou percepção de oportunidades.

2.6 Curva de Tuckman (1965)

"Custos existem para serem diminuídos, não calculados."
– Taiichi Ohno

A teoria proposta por Tuckman em 1965 apresentava 4 fases para um processo de formação de grupos (times) – *Forming*, *Storming*, *Norming* e *Performing* – tendo incluído uma nova etapa anos depois, a que chamou de *Adjourning*. A evolução não é linear, pois conforme saída ou entrada de integrantes, crises ou mudanças, podemos retroceder ou avançar.

Essa teoria é 100% aderente aos princípios e métodos ágeis, voltada a pequenos grupos (equipes), já que valoriza a auto-organização, o papel de facilitação, ciclos de melhoria contínua, passíveis de retrocessos e avanços devido à natureza dinâmica de grupos humanos. Cada grupo construirá sua história, não havendo tempos esperados para cada fase.

- **Forming**: é o momento de formação, que carece de tempo para obter integração e sinergia. É necessário alguém que introduza regramentos iniciais, que imprima um ritmo até que todos comecem a perceber-se como um grupo. Aqui iniciamos os princípios que comporão nossa microcultura e valores;

- **Storming**: após o primeiro momento, passamos a gradualmente ganhar sinergia. O até então gestor cada vez mais atua no papel de coach, orientando cada membro que assuma seu papel frente ao grupo. Esta fase pode gerar conflitos enquanto cada um estabelece seu espaço;

- **Norming**: o time ganha coesão e sinergia, com seus integrantes cientes da importância de seu papel e relevância de sua produção para os resultados do grupo, o até então coach passa a atuar mais como um facilitador. O grupo já possui personalidade própria, produtividade, sustentabilidade e autonomia.

- **Performing**: o grupo ou equipe atinge maturidade no desenvolvimento de seus objetivos e torna desnecessária a presença de um gestor, líder, coach ou facilitador. A tônica é auto-organização, senso de unidade e pertença.

- **Adjourning**: a nova etapa incluída por Tuckman diz respeito ao encerramento de um projeto. O papel do gestor volta a ser necessário para reorganização, reorientação e endereçamentos.

A formação de um grupo é essencial e ao mesmo tempo a tarefa mais difícil. Há equipes que se mantêm indefinidamente entre o *forming* e o *storming*, com causas que vão desde os valores e cultura organizacionais até aspectos individuais.

Cabe ao gestor perceber as características do projeto, da equipe, de seus integrantes e trabalhar para uma cultura própria de bem-estar, produtividade, qualidade e inovação. Se não houver a iniciativa de buscar argumentos e percepções na psicologia, sociologia e nas ciências sociais, tudo ficará mais difícil.

2.7 Teoria da agência (1972)

"O coração do sistema é o envolvimento: membros de equipe flexíveis e motivados, constantemente à procura de uma forma melhor de fazer as coisas."
– Pascal Dennis

Alchian e Demsetz teorizaram sobre a necessidade de os proprietários contratarem agentes, que são os executivos que lhes representam para a execução da estratégia junto aos diferentes níveis da organização, visando garantir os desdobramentos estratégicos, execução tática e coordenação operacional segundo sua vontade e visão.

Após a revolução industrial e a produção em massa instigarem o empreendedorismo em grandes empresas, criou-se a figura dos grandes proprietários dos meios de produção, grandes organizações que, à luz das novas teorias da administração, constituíam hierarquias em suas estruturas. Conselhos, diretores, gerentes, coordenadores, supervisores e operários, modelos que evoluíram com o passar das décadas, mas que guardam em seu bojo gargalos que terão que ser desfeitos para podermos dar o próximo passo.

Não era mais possível ao proprietário acompanhar a execução da estratégia, menos ainda dos desdobramentos táticos. Por isso, passaram a contratar seus agentes (diretores), que por suas vezes contratavam e delegavam a seus gerentes, que selecionavam seus coordenadores, de forma que uma linha única conduzisse o fluxo decisório desde o topo até a base da pirâmide organizacional.

Um dos conflitos estudados pela teoria é o de interesses. O agente tem restrições quanto a assumir riscos que prejudiquem sua carreira, logo, trabalhará para garanti-la em curso ascendente, mesmo com algumas liberdades inconfessas em detrimento daquilo que seria o melhor em médio ou longo prazo para a organização.

2.8 Teoria institucional (1977)

"Aplicar o Just In Time (JIT) sem minimizar o desperdício verdadeiramente não é aplicar o JIT."
– Kaizen, Masaaki Imai

Criada por Meyer e impulsionada em 1983 por DiMaggio e Powell, a Teoria institucional descortina o debate sobre o isomorfismo organizacional, ou seja, quando uma empresa copia modelo, processo ou aspectos de outra para obter maior visibilidade, competitividade, legitimidade ou unidade frente a seu campo organizacional.

Um campo organizacional é formado por um grupo de organizações que, juntas, constituem uma área reconhecida da vida institucional, podendo ser percebido entre parceiros, concorrentes, cadeias produtivas etc. Os quatro elos desta corrente são: interação, dominação, informação e pertença.

As principais forças que as organizações devem levar em consideração são as outras organizações (ALDRICH), elas não competem somente por recursos e clientes, mas por poder político, legitimação institucional, por adequação social, assim como por adequação econômica.

- **Isomorfismo Competitivo** resume-se a assemelhar-se por oportunismo, por questões de mercado, na busca por nichos ou aptidões. Ações comuns em mercados mais competitivos;

- **Isomorfismo Institucional Coercitivo** cede a pressões formais ou informais, explícitas ou sutis, por persuasão ou trama, governamental ou no anseio por aceitação ou legitimidade;

- **Isomorfismo Institucional Normativo** é baseado na profissionalização, no cognitivo, congressos e eventos, regulamentação e legitimação, na contratação de pessoas com currículo semelhante;

- **Isomorfismo Institucional Mimético** é o mais curioso, motivado por incertezas, metas ambíguas, tecnologias insuficientes ou benchmarks. Mimetismo pode ocorrer mesmo em dissonância aos valores da empresa, gerando oportunidades, mas também riscos.

Quão importante é sabermos por que estamos adotando métodos ágeis, open space, entendermos realmente os objetivos? Será que está claro a todos? Você está estudando e adotando métodos ágeis por quê? A cultura da sua organização será aliada ou é uma restrição nesta caminhada?

2.9 Teoria das restrições (1984)

"Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados."
– Mahatma Gandhi

A Teoria das restrições (TOC) é uma metodologia de gestão idealizada pelo Dr. Eliyahu Goldratt, físico, cientista, educador e líder de negócio. É uma filosofia de negócios que se utiliza de princípios científicos. Muitos efeitos são explicados por poucas causas; devemos conhecer e resolver as causas em vez de gerar desperdício atuando em seus efeitos:

- **Identificação** – Identificar a principal restrição (causa);
- **Exploração** – Investir no aperfeiçoamento desta causa;
- **Subordinação** – Analisar os processos relacionados;
- **Elevação** – Se não surtir efeito, reaperfeiçoar;
- **Repetição** – Ao eliminar uma restrição, reiniciar.

A sugestão é termos uma visão de causas e efeitos. O médico analisa sintomas e quadro clínico para emitir um diagnóstico de causa: a dengue (causa) gera efeitos como febre, dor de cabeça e mal estar (efeitos); tratar separadamente os resultados será insatisfatório, mas se solucionarmos a causa, todos os efeitos desaparecerão.

Se a empresa é um sistema, como os elos de uma corrente, não adianta fortalecer indiscriminadamente cada um destes elos. Devemos permanentemente verificar qual destes elos é o mais fraco, qual está comprometendo o resultado final, logo, é este que deve ser melhorado.

O nome que a TOC dá ao elo mais fraco da corrente é Restrição. Todos os sistemas sempre possuirão restrições. Atuamos para identificá-las e trabalhamos na tentativa de resolvê-las ou mitigá-las. Mas sempre existe o risco de pressupor incorretamente e novas tentativas serão percebidas em um ciclo de melhoria contínua.

Os sistemas puxados (*pull system*) são baseados nos mesmos fundamentos da TOC. A produção puxada controla as operações sem a utilização de estoque em processo, valoriza um fluxo contínuo de construção e outros pilares da filosofia Lean.

2.10 Matriz Cynefin (1999)

"O problema não é que existem problemas. O problema é esperar que seja de outra forma e pensar que ter problemas é um problema."
– Theodore Rubin

A matriz Cynefin de David Snowden apresenta quatro tipos de sistemas. Ela nos ajuda a entender que em desenvolvimento de software o caminho é incerto, complexo e imprevisível. Exige conhecimento, criatividade, interação e um tanto de empreendedorismo.

Em software, o modelo mecanicista em que tudo deve estar previsto e deve seguido é fantasioso. Ao tentarmos negar a incerteza, apenas aumentamos os riscos, pois lidamos diariamente com o erro. Sistemas complexos exigem a criação de ambientes criativos, que privilegiem a transparência e a comunicação, a interação e adaptabilidade. Ciclos iterativos incrementais não eliminam, mas antecipam os riscos.

- **Simples** – As relações de causa e efeito são repetitivas, visíveis e previsíveis, o domínio é conhecido e baseado em melhores práticas. É possível planejar e executar conforme o planejado. É só planejar e seguir o plano.

- **Complicado** – As relações de causa e efeito repetem-se, mas nem sempre de forma previsível. O domínio é provável, baseado em boas práticas. É possível planejar e durante a execução buscar soluções já mapeadas;

- **Complexo** – As relações de causa e efeito frequentemente não se repetem, nem são previsíveis. O domínio sugere práticas adaptativas, emergentes. É possível estimar, mas durante a execução novas soluções deverão ser mapeadas;
- **Caótico** – As relações de causa e efeito não são percebidas, bem como existem padrões desconhecidos. O domínio exige novas práticas, decisões são tomadas sem mapeamentos prévios.

2.11 Gamification (2002)

"A estratégia deve ser barata; o aumento da produtividade deve ser feito sem investimentos significativos. Não se deve aplicar somas astronômicas em tecnologias e consultorias."
– Kaizen, Masaaki Imai

Nas raízes da *Gamification* de Nick Pelling e no Agile, há a transformação de ambientes e produtos, mas não se trata de fazer algo divertido e voltar para a rotina, isso é descompressão. Não é exercitar-se, isso é laboral. Não é um momento de dispersão bancado pelo chefe em meio à pressão desmedida, isso é dissimulação. Não é usar Agile sem nada melhorar para as pessoas e negócio, isso é desperdício.

Gamification pode ser um poderoso aliado aos princípios ágeis na construção de um modelo mental orientado a jogos para engajar, recompensar, estimular, treinar e inovar. O uso de conceitos de jogos visa transformar a forma de ver e executar o trabalho em algo mais envolvente, menos repetitivo, mais sustentável, com inovação e empreendedorismo.

Agile Thinking e *Game Thinking* são modelos mentais, uma visão holística que nos desafia a sermos mais sustentáveis, *Human Thinking* afinal. Mesmo as empresas que não estão dispostas a tentar podem mitigar os efeitos de uma cultura difícil através de um mínimo de valorização das suas pessoas, melhorando o convívio e a comunicação e tentando reter seus talentos.

- **Localização** – Como tudo mais, é preciso dar atenção aos aspectos psicológicos, valor do design, preparação do ambiente, atividades aplicadas de forma variada, divertida e instigante. Tendo alguma opção, é

bom usar lugares diferentes do dia a dia: quando está calor, sugerem-se locais ao ar livre, terraços, sob copa de árvores. Surpreender é bom e gera uma vibe positiva.

- **Materialização** – Uso de agile games com chapéus, equipamentos, personagens, medalhas, avatares, pontuações, cores, identidade, torneios. Há fotos de treinamento em que as pessoas usam chapéus divertidos, luvas, jalecos, colares com personagens animados, avatares divertidos, post-its customizados e cartões coloridos.
- **Amplificação** – Muitas vezes é possível prolongar as experiências através de artefatos, campanhas, tarefas prévias ou posteriores de forma a manter a mobilização e potencializar os efeitos. Um exemplo são eventos em que as pessoas leem um texto e trazem suas impressões ou provocações relativas àquela abordagem ou texto.

Capítulo 3

Princípios ágeis

Quais são os princípios que norteiam e quais os pilares que sustentam os métodos Agile? O que é afinal o exercício destes valores e premissas? É fundamental que cada profissional tenha a noção exata de que este caminho não é fácil. Para obter sucesso em metodologias ágeis é preciso que todos se esforcem em crescer coletivamente, apoiando uns aos outros para que todos sejam mais que o somatório de suas individualidades.

3.1 Por que o método se chama "Ágil"?

"Uma reunião em que todos os presentes estão absolutamente de acordo é uma reunião perdida."
– ALBERT EINSTEIN

O termo "ágil" gera uma confusão, pois algumas pessoas entendem que a construção será mais rápida. A referência é relativa à antecipação de valor, oportunidades e riscos, enquanto a tendência do tempo total tende a ser mais longa, mas garantindo qualidade nos resultados empresariais. A seguir uma alegoria, categórica a título de ilustração:

- A construção tradicional levanta as colunas e vigas, lajes, sobe até o último andar e, enquanto isso, ergue as paredes, dutos, eletricidade, hidráulica o mais rápido que podem. O foco é fazer tudo em menos tempo possível, com o menor custo e menor desperdício possível de material, ganho de escala, apenas com especialistas, cada um no seu quadrado (= 18 meses).
- A construção em modelo ágil quase interrompe no terceiro mês para montar o apartamento decorado, completo, com refinada decoração, preparação de parte do hall e áreas comuns, em especial o que estiver no caminho, um gazebo, jardim e hall que chame a atenção dos *prospects* e clientes, deixando visível o potencial do empreendimento (= 20 meses).

Os 20 meses são mais ágeis que 18, porque sabemos e atendemos o objetivo real, que não é construir rápido, é oferecer algo diferenciado: não é só vender apartamentos, mas vender o sonho do melhor local para viver a vida. Compartilha-se a entrega antecipada de um apartamento real, decorado, e permite à própria empresa e a seu público interagir, tirar suas dúvidas, entender, antecipar melhorias, propor extras que agreguem valor ao resultado final.

O manifesto ágil fala em entrega antecipada de valor, de forma iterativa, em entender e satisfazer seu cliente, com qualidade, corrigindo erros o quanto antes etc. Isso tudo fica mais fácil com o apartamento decorado e, com certeza, teremos assim mais vendas desde o primeiro semestre até o final da obra, satisfazendo a todos os envolvidos.

Isso é a essência da agilidade, transformar as percepções em fatos, perceber problemas antecipadamente, buscando, neste processo, agregar valor ao maior número de participantes e interessados, bem como garantir os resultados e promover satisfação.

3.2 Glanuralidade ágil

"Focaliza a atenção no Genba, local onde se cria realmente o valor."
– Kaizen, Masaaki Imai

Uma provocação pelos métodos ágeis é a necessidade de quebrar o que se quer em pedaços menores, tanto quanto possível, até chegar à escala de horas, facilitando a estimativa prévia e a entrega contínua de algo de valor:

- **Visão/Missão** – Tudo começa com uma ideia e missão;
- **Estratégia** – É uma das disciplinas mais importantes. Saber o que é valor, decidir quais soluções são realmente necessárias, entender qual é a visão, a cadeia de valor, qual a direção e objetivos;
- **Release** – Um planejamento de médio prazo também é muito importante, cada release (projeto) deve ter objetivos claros de valor que serão agregados ao atingi-los, equilibrando desejos e superpoderes aos diferentes interessados.
- **Iteração ou Sprint** – São períodos curtos, entre 2 e 4 semanas, destinados à construção do SW desejado. Cada Sprint possui eventos que lhe materializam – *planning*, *daily*, *review*, retrospectiva.
- **Diária** – Diariamente, no mesmo local e hora a equipe se reúne em pé, junto ao quadro de tarefas, e em 15 minutos gera um status atualizado com foco em oportunidades, impedimentos e desvios, pelo sucesso do Sprint.
- **Qualidade/Engenharia** – Em cada jornada de trabalho temos a construção de software de qualidade, com boas práticas de engenharia, visando não gerar dívida técnica, com *pair programming*, *review* e testes automatizados.

3.3 Manifesto Ágil

"O produto final do planejamento não é a informação: é sempre o trabalho."
– Peter Drucker

Em fevereiro de 2001, em uma estação de esqui em Utah, EUA, 17 profissionais que já vinham praticando novos métodos de trabalho em desenvolvimento de software, publicando e divulgando metodologias rotuladas como leves (*lightwave*), se reuniram para declarar os pontos em comum que estavam chamando a atenção do mercado pelos bons resultados que vinham conquistando, por exemplo:

- 1992, Crystal Clear Method – Alistair Cockburn;
- 1993, Scrum – Shuterland, Schwaber e Beedle;
- 1994, Patterns, UML, XP – Martin Fowler;
- 1996, XP – Beck, Cunningham e Jeffries;
- 1997, FDD – Jeff De Luca e Peter Coad;
- 1997, ASD – Highsmith e Cockburn.

A seguir, alguns pontos comuns a estes métodos: times pequenos, auto-organizados, menos hierarquia e mais autonomia; foco em valor, o que realmente faz a diferença para o negócio e pessoas; ciclos curtos, iterativo-incrementais, na escala de semanas; busca pela qualidade consciente, em valor sem desperdícios; liberdade com responsabilidade, devemos ter orgulho da equipe e produto; gestão visual, realismo e transparência, tomando decisões diariamente; usar uma linguagem ubíqua, comum junto a todos os envolvidos.

3.4 Princípios ágeis

"Prioridade às pessoas, esforço principal de melhoria deve vir de uma nova mentalidade e estilo de trabalho das pessoas para a qualidade, equipe, sabedoria, elevação da moral, autodisciplina, círculos da qualidade e prática de sugestões individuais ou de grupo."
– Kaizen, Masaaki Imai

O manifesto para o desenvolvimento ágil de software possui quatro artigos e doze princípios, com o seguinte texto:

"Estamos descobrindo maneiras melhores de desenvolver software fazendo-o nós mesmos e ajudando outros a fazê-lo. Através deste trabalho, passamos a valorizar: indivíduos e interação entre eles mais que processos e ferramentas; software em funcionamento mais que documentação abrangente; colaboração com o cliente mais que negociação de contratos; responder a mudanças mais que seguir um plano."

Eis a relação dos princípios por trás do manifesto ágil, experimente perceber o quanto você acredita e o quanto você segue cada um deles:

1) Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor;

2) Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas;

3) Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos;

4) Pessoas relacionadas a negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto;

5) Construir projetos ao redor de indivíduos motivados, dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho;

6) O método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara;

7) Software funcional é a medida primária de progresso;

8) Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter, indefinidamente, passos constantes;

9) Contínua atenção a excelência técnica e bom design aumentam a agilidade;

10) Simplicidade, a arte de maximizar a quantidade de trabalho que não precisou ser feito;

11) As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis;

12) Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajusta e otimiza seu comportamento de acordo.

3.5 Algumas pessoas olham para o lado errado

"A ideia é que a empresa tente absorver a criatividade e a capacidade de solucionar problemas de todos os seus funcionários."
– Michael Hoseus

Nenhuma empresa é obrigada a adotar Agile e nitidamente algumas o fazem por simples mimetismo ou marketing, pois continuam acreditando em silos, em projetos compostos por um ecossistema compartimentado. É como uma prefeitura que divulga sua sustentabilidade, mas despeja esgoto direto no rio, desperdiça energia e água potável etc.

Tive a oportunidade de participar de projetos de sucesso, sem termos nada de excepcional. Éramos todos imaturos no método, tínhamos conflitos e idiossincrasias. A arquitetura e engenharia tinham o que melhorar, mas deu certo porque tínhamos áreas de negócio e corporativas tão ágeis quanto nós, linguagem ubíqua, e todos tentavam entender o que era valor e desperdício!

Sucesso ágil não é ter um produto "bonito" que não rentabiliza, não é chegar aos resultados a qualquer custo, nem ter um ambiente cool sem rumo, menos ainda ter uma equipe perita no improviso, um discurso sustentável, mas na busca de bodes expiatórios a cada tropeço, eu versus vocês, nós versus eles.

É difícil obter ganhos sem integrar negócio, corporativo e tecnologia, pensando apenas nos objetivos e resultados. Integrar fornecedores, clientes e usuários, pela evolução e rentabilização não é mito, nem tampouco impossível, mas depende de todos comprarem a briga.

Faça workshops Scrum para os times Scrum, mas é fundamental que façamos também workshops e treinamentos de alto nível para times de negócios,

backoffice, fornecedores. Todos os envolvidos devem entender os porquês, ganhos e custos. Somos um elo da cadeia e, se temos uma corrente fraca, reforçar apenas um elo não basta: ao puxar, vai abrir em outro ponto.

3.6 Gestão do tempo

"O lema essencial da aprendizagem organizacional é aprender fazendo."
– Kaizen, Masaaki Imai

Gestão do seu tempo é mais que uma premissa ágil, é caminho para o autoconhecimento. Afinal, onde vão parar as horas de cada dia? Todos se surpreendem ao usar técnicas simples para entender o porquê das frases:

- Não tenho tempo para nada.
- Falta só um pouquinho.
- Não fiz porque estava atolado.
- Já vou dar um jeito, volto em 1 hora.
- Hoje vou ter que ficar até mais tarde.

Em workshops, desafio as pessoas a monitorarem seu tempo por alguns dias. O objetivo não é fazer ninguém trabalhar mais, mas garantir que no fim você será mais consciente e feliz, com mais argumentos para antecipar-se.

Invisto mais tempo naquilo que é mais importante? O que não deu para fazer é de fato aquilo com menos prioridade? Quanto do que eu faço gera valor e quanto pode ser desperdício? O quanto compartilho e valido estas hipóteses com os colegas? Será que pequenas coisas desnecessárias, somadas, não impactam?

Seja consciente, antecipe, organize, inicie cada dia com uma lista de tarefas e as mantenha priorizadas de forma a não iniciar pelas menos importantes. Há matrizes individuais ou colaborativas usadas para a transparência dos tempos realizados, é só escolher a que mais lhe agrada.

Temos hábitos que nem percebemos e nos consomem um tempo que depois nos falta. Algumas tarefas urgentes concorrem com as de baixa importância etc. Tem gente que confunde disciplina com cobrança. Não devemos ser escravos do tempo, mas dominá-lo, entender onde, quando e por quê.

3.7 Estratégias de adoção

"Não basta conquistar a sabedoria, é preciso usá-la."
– Cícero

Como todo processo de mudança organizacional, o processo de adoção pode ir a favor dos ventos proporcionados pela cultura vigente ou contra, pode contar com o apoio do coordenador, gerência e direção, ou tê-los como restrição; mas, mesmo com obstáculos, a adoção de técnicas ágeis gera melhorias, se não para todo o ecossistema como desejado, ao menos internamente para a equipe, a título de comunicação. Existem as estratégias:

Pirâmide

A estratégia mais ortodoxa sugere que selecionemos uma equipe que esteja iniciando um pequeno projeto, com integrantes que potencializem a experiência. Após sua conclusão seria possível dividir a galera entre outras equipes, gerando uma progressão geométrica.

Coach

Essa mesma estratégia (pirâmide) pode ser precedida pela preparação de um bom *Scrum Master*, alguém que orientará a primeira equipe e que, na sequência, poderá iniciar outras equipes, treinando não só o time como outros Scrum Masters, orientando a organização.

Big Bang

Eu e uma colega participamos de um processo ousado, treinamos de forma gradual e sem rupturas mais de 70 pessoas, pertencentes a sete equipes de desenvolvimento de produtos digitais. De forma iterativa, introduzimos em três meses primeiro os princípios e atitudes por trás de uma equipe ágil,

depois a *timebox* de retrospectiva e quadros de gestão visual, e finalmente o ciclo completo Scrum.

Subversivo

Se sua chefia e empresa não entenderam o que é Agile e não autorizam a mudança, é possível negociar algumas técnicas, como retrospectivas e gestão visual. Mesmo sem requisitos ágeis e ciclos iterativos, é possível mostrar o valor em melhorar a comunicação, transparência e adaptação.

Waterfall

Tem empresas que são pouco ágeis na sua essência e ficam debatendo gerencialmente esta opção durante meses ou anos, trazendo cursos e consultores, na expectativa de ter a certeza de que dará certo. Normalmente trabalham em um processo *top-down* e querem estar preparados antes de iniciá-lo. Alguns entram em postergação indefinida.

Atenção

Não há garantias, não importa se optamos por iniciar com 100% do Scrum em uma equipe piloto e formadora de opinião ou baby steps com várias equipes, o segredo está em buscar um mínimo de apoio, botar a mão na massa, e evitar gerar somente expectativas com um estoque de pressupostos, incógnitas, planos e idealizações que podem ser frustrantes.

3.8 Pacto de equipe

"Onde quer que haja tendência para aprender, há processos autocorretivos, com mudanças de hábito; onde quer que haja ação guiada por um propósito, aí há inteligência."
– Lúcia Santaella

O primeiro passo é um "pacto" de equipe, quando em conjunto listamos tudo o que o time julga necessário para uma convivência saudável. O quadro

tem só duas colunas: "O que queremos" e "O que NÃO queremos." No final, ele é colocado na parede, à vista, sendo atualizado sempre que necessário. Normalmente comprovamos a Hierarquia de necessidades de Maslow. Há exceções, mas as necessidades humanas são como uma pirâmide, seguindo uma ordem inconsciente, da base para o topo, primeiro fisiológicas, depois de segurança, social, autoestima, relacionamento, estima e realização. As listas devem ser espontâneas, veja a seguir quatro exemplos:

"Queremos": SER OUVIDOS! É ser respeitado e ter a atenção devida quando falando, é ter retorno sobre o assunto encaminhado, é não ter um interlocutor respondendo e-mails ao mesmo tempo em que diz estar lhe ouvindo;

"NÃO queremos": PERTURBAÇÃO DO AMBIENTE! É não discutir em voz alta questões pessoais no celular, a relação com a namorada, gargalhadas vendo o Youtube, xingamentos, ter mais respeito pelos colegas;

"Queremos": ENTENDER O PORQUÊ! Porque sim não é resposta, transparência é o principal pilar de sustentação do Scrum e precisamos saber por que algo é valor, desperdício, prioridade, cancelamento, postergação etc.;

"NÃO queremos": LEVAR PARA O LADO PESSOAL! É não ficar magoado com qualquer crítica ou comentário dito para ser construtivo. Responda, não silencie e guarde mágoa.

Capítulo 4

Introdução ao método

Por onde começar a entender o Scrum? Como tudo começou? Quem são os maiores nomes, qual o espírito, quais são os fundamentos essenciais? Como chegamos à construção de produtos com alto valor agregado, com o mínimo de desperdício, ao mesmo tempo em que desenvolvemos o espírito de equipe e senso de pertença em todo o ecossistema?

4.1 Scrum

"Scrum humaniza o desenvolvimento de produtos através da introdução de uma comunicação regular, ajudando equipes a se comprometerem com metas compartilhadas."
– Ken Schwaber e Jeff Shuterland

O termo Scrum vem do jogo de rugby, é a jogada em que ficam todos juntos, cada um apoiando os demais, frente a frente com time adversário. Acontece sempre que a bola para ou sai de campo. Cada "scrum" força que os times se auto-organizem e reiniciem o jogo.

A primeira referência foi em um artigo de Hirotaka Takeuchi e Ikujiro Nonaka em uma publicação da Harvard Business Review de 1986, chamada *The New New Product Development Game* apresentando os conceitos e fundamentos de times ágeis, pequenos, multidisciplinares, auto-organizados, com foco em entrega de valor e melhoria contínua.

Em 1995, Jeff Sutherland e Ken Schwaber uniram esforços para a formalização do método que chamaram de Scrum, apresentado no artigo *Scrum and the Perfect Storm*. Jeff era vice-presidente de engenharia na Easel e Ken trabalhava na Advanced Development Methods. Ambos estavam presentes quando da assinatura do Manifesto Ágil em fevereiro de 2001.

Desde então, Schwaber e Sutherland vêm lançando atualizações do documento batizado por eles de *Scrum Guide*, que contém na sua versão 2011 tão somente dezoito páginas, com os papéis, timeboxes, artefatos e regras, leitura obrigatória para quem quer começar a rodar: http://www.scrumguides.org.

Outra informação pertinente é o entendimento de que estes dois grandes nomes do Scrum acabaram fundando e apoiando as duas instituições certificadoras do método, a http://scrumalliance.com e http://scrum.org.

4.2 Os três pilares do Scrum

"O primeiro passo indispensável para conseguir as coisas que você quer da vida é este: Decida o que você quer."
– Ben Stein

Cada momento do método Scrum para desenvolvimento de software está baseado em três pilares recorrentes, que são permanentemente praticados:

- **Transparência** – Para saber se estamos no caminho certo, é preciso que todos se posicionem, diariamente, com sentimento de pertencimento; o projeto é de todos;

- **Inspeção** – Identificada uma oportunidade ou risco, todos têm a obrigação de fazer o seu melhor e buscar o melhor do time a cada relato ou participação; todos colaboram;
- **Adaptação** – Identificada a necessidade ou oportunidade de um plano de ação, todos participam e empenham-se pelo sucesso de todos.

Esse método garante que todos possam falar e ser ouvidos, com realismo, educação e cooperação permanente. Se você leu todos os capítulos antes de chegar neste ponto, já sabe que não é fácil, pois as pessoas tendem a ser individualistas e protecionistas.

Fundamentos dos três pilares

Em um *mindset* ágil evitamos fazer surpresas, não importa se para algo novo ou um problema, bom ou ruim. O quanto antes todos souberem, mais podem contribuir. Ideias evoluem se forem colaborativas, riscos são mitigados quando todos estão empenhados em diminuí-los.

Outro conceito fundamental é o significado do termo japonês "*Genba*", que significa haver um lugar e hora certa para falar as coisas. Não fale pelas costas, não minta nem diga meias verdades, mas seja cortês, educado e objetivo.

A transparência é inimiga mortal de atitudes prolixas. Devemos aprender a falar objetivamente, de forma assertiva, é uma habilidade que devemos desenvolver, bem como o poder de argumentação: alguns ficam magoados por não convencerem alguém, mas não percebem que mais depende da força e validade dos seus argumentos.

4.3 Overview do método Scrum

"Escolher o seu tempo é ganhar tempo."
– Francis Bacon

O framework Scrum possui um fluxo iterativo-incremental, o que quer dizer que trabalhamos em camadas. Permanentemente estamos iniciando uma nova, construindo um pedacinho do todo, aquilo que de mais relevante

houver; entregamos, validamos e então reiniciamos na escala de dias ou semanas. Temos reuniões diárias, dentro de ciclos semanais, em uma estratégia mensal para a construção de um negócio, produto ou serviço vencedor.

Fig. 4.1: Método Scrum

Dois princípios relevantes a cada um destes ciclos que chamamos de Sprints é trabalhar em pequenas porções e exercitar entregas frequentes, com valor e qualidade, daqueles fragmentos mais importantes. Assim, o negócio constantemente afere se estamos no caminho certo, um passo de cada vez, postergando a decisão do restante para quando realmente for preciso.

Iniciamos pela **Visão**, nossa missão, macro-objetivos e contextualização, os porquês de este produto existir ou ser construído. Para fazê-lo, precisamos montar e manter um **Product backlog**, que é uma lista de requisitos mínimos ou que agreguem valor ao produto.

Com estratégia definida e backlog priorizado, fica mais fácil selecionar para cada ciclo de desenvolvimento o seu **Sprint backlog**, uma lista de requisitos desejados para o ciclo das próximas semanas, e então estimar cada pedacinho através do **Sprint planning**. Isso é quando a equipe estima cada tarefa necessária para este objetivo de curto prazo e faz um pacto para o sucesso desta entrega.

Chamamos de **Sprint** o ciclo composto desde o Sprint planning até a entrega do resultado de 2 a 4 semanas de trabalho. A cada 24 horas através da **Daily scrum meeting**, nossas reuniões em pé, cada integrante do time se posiciona quanto ao seu trabalho para o sucesso daquilo que estamos construindo.

No último dia do tempo previsto para o Sprint ocorre a **Review**, uma reunião do time com seus *stakeholders*. O objetivo é apresentar e discutir o que foi feito, realinhar expectativas e pressupostos à luz das partes concluídas e entregues. Finalmente, fazemos a **Retrospectiva**, momento de reflexão do time quanto ao andamento do seu trabalho, o que melhorar, manter ou crescer.

E quanto à entrega daquilo que foi construído? Todo o ciclo de execução de um Sprint foca em entregar valor pronto, um *potentially shippable product increment*. Mas não é incomum que a publicação em produção dependa de autorização do negócio, que às vezes envolve questões publicitárias, comerciais, parceiros ou tão somente estratégia de lançamento. É importante que o time esteja a par de forma a estimar sua participação direta ou indireta quando do lançamento.

O segredo do sucesso de métodos ágeis é o equilíbrio entre as exigências do negócio e a sustentabilidade, entre o valor e expectativas com a qualidade entregue, com ou sem dívidas técnicas. Seguindo um plano formal ou inovando a cada passo, sem equilíbrio entre estratégia, processo, pessoas, engenharia, o resultado será predatório para um destes pilares. A seguir apresento um breve glossário ágil introdutório:

1) **Papéis**: o Product Owner como sendo o analista de negócios e representante do cliente; o Scrum Master para suporte ao método e cadência; a Equipe de desenvolvimento para construção e valor;

2) **Product Owner**: é o papel de maior visibilidade, o navegador do time, é ele quem toma as decisões estratégicas, define o que precisa ser feito, valida e traz do negócio a palavra final para aceite e publicação;

3) **Scrum Master**: um profundo conhecedor do método e técnicas, proporcionando treinamentos e reciclagens, organização dos eventos, desimpedimentos, e trabalhando pela harmonia do time em seu ecossistema;

4) **Equipe Scrum**: contém os cargos necessários à construção, como UX, SEO, arquiteto, desenvolvedor, testador, entre outros, sempre valorizando certo nível de multidisciplinaridade;

5) **Linguagem ubíqua**: este termo traduz à perfeição o uso de uma linguagem comum aceita e entendida pelo usuário e demais envolvidos para a

descrição do que deve ser feito, deixando o "tecniquês" à equipe;

6) **User story**: e uma unidade de documentação que declara cada um dos requisitos da solução desejada, escritos pela perspectiva dos usuários ou stakeholder (envolvidos), indicando quem quer, o que e o porquê.

7) **Timeboxes**: o Scrum é cadenciado pela existência de eventos com periodicidade e tempos de duração preestabelecidos, garantindo ciclos PDCL (*Plan-Do-Check-Learn*), para entendimento, construção, entrega e retroalimentação;

8) **Visão**: há técnicas ágeis de modelagem de negócios e produtos que permitem ao Product Owner entender e mapear a missão, objetivos e prioridades de forma a diminuir o desperdício e agregar valor a cada ciclo;

9) **Product backlog**: é a lista de demandas, sonhos e oportunidades pendentes gerenciada pelo Product Owner, sempre atualizada de forma a registrar todas as solicitações e necessidades do usuário e técnicas;

10) **Release planning**: vejamos a Release como sendo um projeto ou fase; há técnicas ágeis para a organização e planejamento de expectativas sobre as principais necessidades contidas no Product backlog;

11) **Sprint**: cada Release é dividida em um ou mais Sprints, que existem para garantir frequência na entrega de valor ao negócio no máximo a cada 2 a 4 semanas, de forma a manter um fluxo de melhoria contínua;

12) **Sprint backlog**: é aquela porção do Product backlog que o Product Owner elegeu como mais importante neste momento e que apresentará ao time para construção e entrega ao negócio no próximo Sprint;

13) **Sprint planning**: na manhã do primeiro dia do Sprint ocorre seu planejamento; a equipe irá assistir a apresentação das User stories pelo Product Owner, estimar e fazer um pacto para construção e entrega;

14) **Daily scrum meeting**: Reunião no mesmo local e horário, todos os dias, com a presença de todos do time. Cada integrante deve exercer os 3 pilares, relatando a sua condição e inspecionando as demais;

15) **Scrum board**: o quadro de tarefas garante visibilidade do status das User stories e tarefas; o mínimo tem 3 colunas (to do ... doing ... done), mas cada equipe cria quantas linhas e colunas forem mais adequadas;

16) **Review**: no início do último dia temos a reunião de Review, com todos os envolvidos convidados para que todo o time apresente o que construiu e como foi o Sprint, a seguir alinhando os próximos passos;

17) **Retrospectiva**: no final do último dia temos a reunião de Retrospectiva; o time se encontra para discutir os detalhes do Sprint, produtividade, problemas, erros e acertos, melhorando seu método e valores;

18) **Entrega**: ao final do Sprint é disponibilizada a entrega daquilo que foi combinado, construído e concluído, entretanto não é incomum a entrada em produção aguardar a melhor data para o negócio;

19) **Fases**: o ciclo de vida do método Scrum é dividido em dois momentos concorrentes, o de Discovery olhando mais à frente, definindo os próximos passos e a estratégia, e a fase de Delivery, para construção e entrega;

20) **Discovery**: etapa composta pelos passos de Visão, Product backlog, Release plan, Sprint backlog e início do Sprint planning. Nela está contida uma infinidade de técnicas e dinâmicas importadas de outros métodos;

21) **Delivery**: etapa desde o Sprint planning até a Retrospectiva, responsável pela construção com qualidade daquele conjunto de requisitos pactuados no Sprint planning e que definem o sucesso do Sprint.

4.4 Product Owner

[quote "Todos na empresa têm de estar envolvidos, desde os gestores do topo e intermédios, até o pessoal de base; a metodologia não é elitista." Kaizen, Masaaki Imai]

O Product Owner é o papel de maior responsabilidade e visibilidade no método. Ele representa o cliente, é responsável pelas decisões sobre qual trabalho maximizará o valor do produto e agregará mais aos envolvidos, decide

metas, prioriza e aprova, sempre buscando garantir o ROI (*Return On Investiment*).

Deve ter cuidado com o sentimento de "posse". Em métodos ágeis, cada integrante deve ter a humildade e orgulho de pertencer ao time. O sucesso ou o insucesso será por igual, de todos.

O que diz o Scrum Guide?

O Product Owner é o responsável por gerenciar o Product backlog do produto, pode pedir a um analista de negócios, SEO ou sistemas, QA ou arquiteto, que lhe auxiliem no entendimento, registro, diagramação do fluxo, mas será sempre o responsável final pelas decisões de negócio junto ao time:

- Registrar com clareza os itens do Backlog do produto;
- Ordenar assertivamente os itens do Backlog do produto;
- Garantir que o Backlog do produto seja visível e claro a todos;
- Garantir que a equipe entenda 100% os itens do Sprint backlog;
- Garantir o valor do trabalho desempenhado pela equipe;
- Perceber o MVP (mínimo produto viável) e melhoria contínua em ciclos iterativos.

Ninguém está autorizado a mandar a equipe de desenvolvimento trabalhar em um conjunto diferente de requisitos ou mudar sua prioridade à revelia do PO. Isto **não** quer dizer que ele não pode ser questionado, mas qualquer mudança no planejado deve ser compartilhada e decidida **com** ele.

Cotidiano de um P.O.

Eu trabalho em um departamento de TI em que o PO é um colega e está disponível a qualquer momento, é diferente de uma empresa prestadora de serviços que trabalha com um PO do cliente, mas em ambos os casos o PO deve dedicar-se a este papel no máximo de tempo.

Digo que 50% do tempo de um PO ele passa sentado junto aos usuários, envolvidos e interessados, e nos outros 50%, junto à equipe, participando das reuniões diárias, planejamento, retrospectivas e reviews. Cuidado, um PO que fica remoto porque tem "mais o que fazer" é a fórmula do insucesso!

O Product Owner não possui hierarquia sobre a equipe enquanto exerce seu papel, sua força é conhecimento e argumentação, mantendo sempre o foco na obtenção do melhor do processo, pessoas e recursos para atingir o máximo de valor para o negócio, de forma equilibrada e sustentável.

Um erro comum de um PO iniciante é continuar usando a conjugação na 1ª pessoa do singular e 3ª do plural, "eu" e "eles", enquanto o único meio de ter uma equipe Scrum é usar só a 1ª do plural, "nós".

Discovery e Delivery

Na etapa de Discovery, desde a Visão até o Sprint planning, seu papel é montar e gerir o Product backlog, extrair dele a melhor definição de Release e de cada Sprint backlog, de forma a sempre oferecer à equipe o melhor entendimento daquilo que mais agregará valor ao produto.

O PO não é cargo de gabinete, deve ir a campo diariamente, lançando mão de eventos que valorizam a participação e convergência, *Business Model Canvas*, *Story mappings*, *Brainstormings*, *Focus Groups*, pesquisas, abrir canais junto aos envolvidos, ampliar oportunidades.

A etapa de Delivery, a partir do Sprint planning até a entrega e retrospectiva, exige permanente entendimento do status da iteração por parte do PO, de forma a diariamente tomar as decisões necessárias para correção de desvios, apoiando a equipe nos planos de ação de contorno e ajustes.

Cuidado com a reserva de mercado

Um erro clássico é o PO manter-se como único elo entre negócio e equipe, interfaceando de tal forma que os usuários desconhecem o restante do time e o time não possui qualquer vínculo com usuários. Esta é uma reserva de mercado perigosa, que centraliza e bitola toda informação.

O Product Owner deve sempre que possível convocar integrantes da equipe (analistas, arquitetos, QA etc.) para reuniões na área de negócios e

para as timeboxes do ciclo de Discovery, garantindo visões complementares e divergentes, ampliando debates e qualificando decisões.

4.5 Scrum master

"O desperdício é o inimigo número 1, para eliminá-lo é preciso sujar as mãos!"
– Kaizen, Masaaki Imai

No método Scrum, cabe ao Scrum Master treinar, orientar e provocar. Ele precisa conhecer a empresa e seus colegas, incentivar o crescimento das pessoas, trabalhar para que cada timebox atinja seus objetivos, para elevar o espírito de time.

O que diz o Scrum Guide?

É do Scrum Master a responsabilidade de manter o time em equilíbrio, dando condições e orientando para que o método e os objetivos sejam seguidos:

- Não exerce chefia, ele orienta, facilita, mas não manda;
- Difusão dos princípios ao time, negócio e empresa;
- Facilitar as timeboxes, papéis, artefatos e regras;
- Treinar novos integrantes e reciclar os antigos;
- Facilitar desimpedimentos que possam afetar o fluxo.

Responsabilidade direta e indiretamente

Se alguém me pergunta: qual o primeiro passo para a adoção? A resposta é uma só, encontre seu Scrum Master, alguém disposto a se reinventar, que acredite nos princípios ágeis, pessoas, coletivo, valor e sustentabilidade, que aceite se expor ao propor e experimentar mudanças.

Se um Scrum Master me pergunta qual o seu primeiro passo, a resposta é: desapegar das suas antigas funções e reservar tempo farto ao estudo, comunidades de prática, eventos ágeis e não ágeis, investir forte em conhecimento.

Ele tem que ter bala na agulha, se não, não o levarão a sério; sua função jamais será fazer mais do mesmo.

Se você gosta da ribalta, fuja deste papel

Scrum Master é uma função de retaguarda, ele não define, programa, testa ou entrega. A medida de seu sucesso é o time produzindo de forma coletiva e com qualidade, timeboxes rolando, sem impedimentos e com objetivos atingidos. Ter feito o dever de casa e não ter chamado a atenção.

Se der tudo certo, beleza, mas se der errado, onde estava o Scrum Master que não percebeu que alguém, o time ou parte do ecossistema precisava de orientação? Por que não alertou e provocou uma reflexão para reação? Mesmo reagindo, se tiver que falar ou interferir com frequência, algo está errado.

Argumentação = conhecimento + crença

Não se aprende a ser Scrum Master com um curso; aprende-se fazendo, ousando, conversando com o mercado, diversificando, buscando inspiração na pedagogia, administração, psicologia, comunicação, lendo e escrevendo muito, debatendo técnicas e vivências, criando e validando pressupostos de como melhorar método e processo.

É dever do Scrum Master incentivar boas práticas, ir a campo, participar de grupos de usuários, comunidades de práticas. Há um arsenal de técnicas para instigar conhecimento tácito e explícito; o dever é aprender e replicar, proporcionar à equipe treinamento, palestras, *lightning talks*, *open spaces*, *fishbowls*, *agile games*, workshops, internos e externos.

Ecossistema

Quando comecei a treinar outras áreas, alguns coordenadores disseram que a equipe estava se desmotivando. Eu perguntava: o que estavam fazendo para reverter esse quadro? Percebi que não haviam designado um Scrum Master, então passei a enfatizar que o coordenador teria que assumir este papel, se não, não daria certo. Para eles aplico workshops sobre os princípios ágeis. É preciso entender seu cotidiano, a abordagem passa a ser um PDCL

ágil. É fantástico construir uma linguagem ubíqua ágil com a galera que não é de TI, ver tesouraria, cobrança, DBM, falando em post-its e retrospectivas com brilho nos olhos.

Scrum não é mais do mesmo e timebox não é reunião

O principal papel de um Scrum Master é adaptar as edições de cada timebox ao time e vice-versa, buscando potencializar, fazer mais, diferente e não do mesmo, melhorar com valor e qualidade. Tem um pouco de cada um e muito do Scrum Master em cada timebox e cada timebox em detalhes é assunto para os próximos capítulos.

4.6 Equipe de desenvolvimento

"Sessenta por cento de todos os problemas administrativos resultam da ineficácia da comunicação."
– Peter Drucker

São todos os profissionais da equipe de desenvolvimento e outros profissionais que participam diretamente. Por exemplo, em produtos digitais também temos colegas de UX e SEO em cada equipe.

Há o interesse e recomendação pela multidisciplinaridade, mas eu vejo esta característica como um meio de eliminar gargalos, e não uma necessidade permanente, pois em software precisamos de especialistas.

- **Analista de SEO e WebAnalytics** – Responsável por otimizar as páginas do site para bom posicionamento nos buscadores e acompanhar a audiência.
- **UX (User Experience Designer)** – Estuda a interface do ponto de vista do ser humano. Especializa-se em ergonomia, usabilidade, informação e navegação.
- **Arquiteto de sistemas** – Uma visão mais profunda em plataformas operacionais, infraestrutura, frameworks e bibliotecas, interoperabilidade, banco de dados, segurança, é mais conceitual e orientação.

- **Programador** – Apesar de o Scrum Guide falar em multidisciplinaridade e ausência de cargos, a prática mostra que cada profissional tem sua especialização, quer backend, frontend, Java, PHP ou Dot NET. Valorizamos quem possa ajudar a desimpedir gargalos, não buscamos generalistas.

- **SQA (Software Quality Assurance)** – Um novo tempo para o processo de qualidade, até hoje trabalhamos com testes pós-desenvolvimento, mas já estamos adiantados em automação, testes de regressão, funcionais etc.

O que diz o Scrum Guide?

A equipe de desenvolvimento consiste de profissionais que realizam o trabalho de entregar uma versão usável que potencialmente incrementa o produto "Pronto" ao final de cada Sprint. Somente integrantes da equipe de desenvolvimento criam incrementos.

As equipes de desenvolvimento são estruturadas e autorizadas pela organização para estruturar e gerenciar seu próprio trabalho. A sinergia resultante aperfeiçoa a eficiência e a eficácia da equipe de desenvolvimento como um todo.

4.7 Fases do Scrum: Discovery x Delivery

"A única certeza do planejamento é que as coisas nunca ocorrem como foram planejadas."
– Lúcio Costa

No Scrum temos uma visão geral, um planejamento de Release em que mapeamos tudo o que os usuários desejam ou necessitam para gerar uma estimativa de complexidade e linha do tempo. O detalhamento acontecerá um pouco de cada vez, o suficiente para as próximas semanas, sempre trabalhando na porção que mais agregará valor ao negócio naquele específico momento.
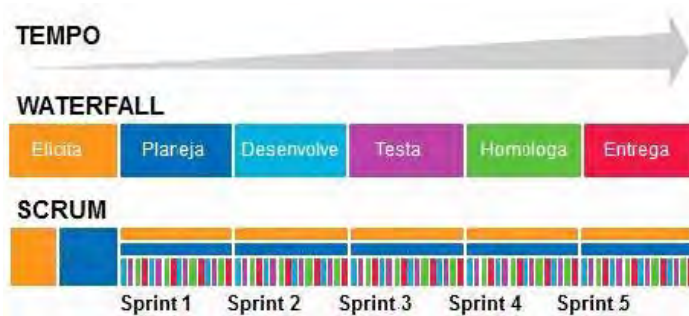
Fig. 4.2: Cascata e iterativo-incremental

Para tanto, é preciso criar um *continuum* em que, sempre ao terminar um Sprint, já tenhamos o detalhamento do próximo para que possa ser estimado e desenvolvido. Isso exige que enquanto um Sprint está sendo construído haja em paralelo um esforço pela definição do próximo.

Assim, o método Scrum trabalha com dois ciclos distintos, já identificados como Discovery e Delivery, que resguardam sua própria natureza e onde diferentes papéis assumem o protagonismo, fato refletido nas características dos quadros de tarefas usados para cada um deles.
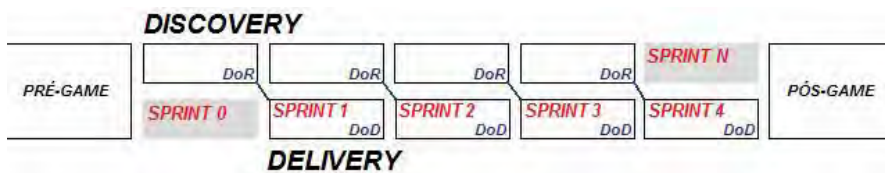
Fig. 4.3: Fases de Discovery e Delivery

Tudo começa pelo *pré-game*, onde é construída uma visão do produto desejado utilizando-se de técnicas colaborativas para elicitar as necessidades reais do cliente e priorizá-las para constituição do Release plan e iniciar o Product backlog.

De posse da lista de funcionalidades priorizadas para o primeiro Sprint, inicia-se o trabalho de detalhamento naquilo que chamamos de ciclos de Dis-

covery. Cada ciclo destes acontece ao mesmo tempo de um Sprint de Delivery, que é quando acontece a construção do software.

No primeiro Discovery temos em paralelo o Sprint Zero, momento único em que a equipe aproveita para as configurações de ambientes, realização de eventuais provas de conceitos e estabelecimento das combinações relacionadas ao cotidiano do trabalho que estão por iniciar.

Cada Discovery possui a responsabilidade de detalhar o suficiente cada uma das funcionalidades priorizadas para o próximo Sprint, de forma que a equipe consiga no início de cada Sprint de Delivery planejar-se o melhor possível para a construção de software de qualidade e valor.

Definition of Ready

O acrônimo dado ao detalhamento necessário de cada funcionalidade durante cada ciclo de Discovery é *DoR*, que identifica a documentação necessária para que cada funcionalidade possa ser considerada e preparada para ser planejada para o próximo Sprint.

Um exemplo possível de DoR combinado é a construção de uma documentação simples onde o próprio usuário especifica o que precisa e por quê, enquanto um analista detalha o protótipo e regras funcionais, de negócios, não de interface.

Definition of Done

O acrônimo usado para que um pedaço de software possa ser considerado apto para ser promovido à produção é *DoD*, que esclarece a todos os interessados quais etapas e práticas devem ser vencidas para que algo desenvolvido possa ser considerado com qualidade e valor para ir para produção.

Um exemplo possível de DoD combinado pelo time é quando uma funcionalidade está desenvolvida, versionada, gerado um *build* e publicado em ambiente de testes, onde terá que ter sido validado com sucesso a partir dos cenários de testes apresentados.

É fundamental que durante o planejamento de cada Sprint seja estimado o tempo de dedicação de alguns integrantes no apoio ao trabalho de Discovery do próximo Sprint, além dos tempos estimados para entender, construir e testar o software previsto para o Delivery do Sprint atual.
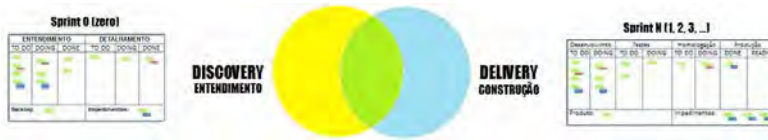
Fig. 4.4: Quadros de Discovery e Delivery

Capítulo 5

Discovery

Projetos e equipes Scrum possuem duas fases imprescindíveis e responsáveis pela sua viabilização e sucesso que são assíncronas e concorrentes. Quer dizer que, enquanto temos alguns papéis e profissionais dedicados a eventos e artefatos para definir os requisitos que serão em breve construídos, a equipe focará em construir o valor que já lhe foi solicitado e pactuado. Apesar de ambas serem vitais, o entendimento do negócio e sua apropriação pelo time é fator de sucesso, vamos iniciar por ele, pelo negócio.

5.1 Visão

"Kaizen significa mudança para melhor, uma filosofia baseada em práticas de melhoria contínua do processo de trabalho."
– Taiichi Ohno

O objetivo da Visão é o esclarecimento da estratégia. Não saberemos o que é valor se não estiver claro quem são nossos clientes, diferencial competitivo, rentabilização, e pelo que seremos cobrados. Um dos erros mais comuns é tomar decisões sem ter alinhado claramente a estratégia. Há diferentes formas de levantar esta discussão, em diferentes patamares: negócio, produto, projeto. O importante é saber qual é e pactuá-lo entre os envolvidos, diretoria, parceiros, áreas envolvidas, time de scrum. No exemplo a seguir, usando a técnica de *Business Model Canvas*:
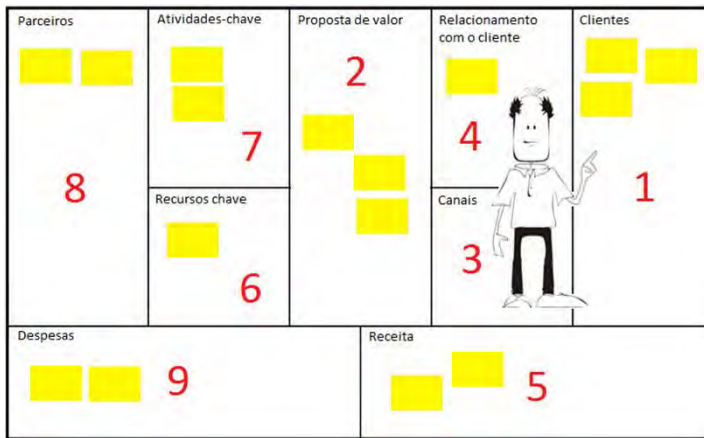
Fig. 5.1: Business Model Canvas

É um artefato vivo que deve ser sempre revisitado:

1) **Segmentos de clientes**: a quem queremos atender, sendo uma lista, a prioridade entre eles, qual o nicho irá melhor direcionar o foco no entendimento de suas características, necessidades, valor.

2) **Proposições de valor**: quais os produtos e serviços que agregariam valor aos nossos clientes, incrementando seus resultados; quais os desejos, poderes, sonhos percebidos e qual a priorização percebida entre eles;

3) **Canais**: especificar os canais de venda, distribuição, comunicação, integração entre eles, buscando o entendimento do potencial de uso de cada um, para o incremento da entrega de valor ao cliente;

4) **Relacionamento com clientes**: qual o tipo de relação a ser estabelecido entre nós e os nossos clientes e/ou usuários, pré e pós-venda, meios, método, custo x benefício, escalabilidade e sustentabilidade;

5) **Fontes de receita**: qual é exatamente a forma de rentabilização possível e esperada, métricas, vantagens e desvantagens, qual o *mindset* de mercado, como o cliente percebe isto e oportunidades de mudança;

6) **Recursos-chave**: recursos que darão a sustentação a este business model, infraestrutura, pessoas, financiamento, aquilo que se fizer necessário para fazer acontecer;

7) **Atividades-chave**: quais são as atividades e ações que sustentarão o modelo na meta de produção, qualidade, vendas, esclarecimento e prioridade das atividades que determinam o atingimento do valor proposto;

8) **Parcerias-chave**: fornecedores e parceiros em função de insumos estratégicos ou materiais que sustentam o modelo e seu sucesso, o existente e o desejado;

9) **Estrutura de custos**: custos diretos e indiretos, fixos e variáveis, qual a priorização entre eles e declaração de riscos, oportunidades e sucesso do negócio.

5.2 User Story

"Poka Yoke significa 'à prova de erros', construir processos ou produtos que minimizem defeitos causados por falhas ou erros humanos."
– Taiichi Ohno

Histórias de usuário (*User story*) são unidades passíveis de serem construídas e entregues, com foco em valor, usadas para documentar as funcionalidades. Devem ser escritas pela perspectiva do usuário, para quem será construída, o que deve ser feito e por que este requisito possui valor.

Como <quem?> eu [quero|preciso|gostaria] <verbo> <quê?> para <por quê?>

- **QUEM?** Cada User story deve, antes de mais nada, indicar quem é o usuário que vai utilizar e por isto tem o maior interesse neste requisito;
- **O QUÊ?** Exatamente o que o interessado quer, precisa ou gostaria que este requisito lhe proporcionasse, pelo prisma de sua interação;
- **POR QUÊ?** Afinal, por que ela existe é a principal informação, aqui temos o valor que estará sendo entregue através desta User story.

Exemplos: (1) Como cliente, quero ver o histórico das minhas aquisições para facilitar minhas decisões; [2] Como gerente de contas preciso um dashboard de informações sobre meus clientes para ser mais proativo.

Detalhar cada interação do usuário segundo o ponto de vista do cliente, quem irá utilizá-la, fracionando-a em situações objetivas para determinado fim. Devem ser simples e claras, passíveis de serem entendidas para serem estimadas pelo time antes de iniciar:

- Independente: devem ser independentes entre si;
- Negociável: não são contratos, mas lembretes;
- Valoráveis: histórias devem agregar valor ao cliente;
- Estimáveis: possíveis de serem mensuradas;
- Testáveis: devem ser possíveis de serem testadas;
- Pequenas: nem grandes nem pequenas demais.

Durante o Sprint Zero, as User stories precisam ser detalhadas, idealizando o design, ergonomia e SEO. Além dos critérios de aceitação que nortearão os planos de testes e validação, inicialmente responsabilidade do Product Owner, a partir do Sprint planning toda a equipe é responsável por manter centralizadas ou linkadas todas as suas definições – negócio, UX, SEO, técnicas.

5.3 Critérios de aceitação

"A revolução da informação representa uma nítida transferência de poder de quem detém o capital para quem detém o conhecimento."
– Peter Drucker

As equipes que já possuem boas práticas de planificação de testes não terão dificuldades em entender e adotar os critérios de aceitação. Trata-se de parte importante de cada User story contendo uma lista de situações que deverão ser validadas para que o Product Owner aceite a entrega como pronta ao final de cada Sprint.

Um aspecto importante é que os testes a serem feitos são um somatório entre os critérios acordados como DoD (*Definition of Done*) e os de aceitação.

Há combinações que pertencem ao produto, Release ou Sprint como um todo e que não precisam estar descritos em cada US, desde questões de arquitetura, formato de data ou critérios básicos por tipo de dado.

Os critérios de aceitação não são peça figurativa, não são uma lista para o testador, menos ainda só para o Product Owner. Há um ciclo de vida:

- O PO lista os critérios de aceitação de cada US;
- A partir do Planning, passa a ser uma construção coletiva, sempre referendado pelo PO;
- O testador os usa como gênese para os casos de testes;
- O desenvolvedor usa os casos de testes para ajudar na construção de testes unitários e valida antes de liberar. Domínio do que se quer e qualidade são obrigação primaz de todos;
- O testador tem responsabilidade muito além dos bugs, Não deveriam chegar ao testador bugs previstos nos critérios e casos de testes;
- O Product Owner deve homologar aspectos funcionais e integrados; não deveriam nunca chegar a ele bugs previstos nos critérios de aceitação e na construção do DoR e DoD.

Há um formato proposto para construção dos critérios utilizado pelo método FDD e que utiliza palavras-chaves para criar cenários e permitir a automação desses critérios usando softwares específicos como Cucumber e JBehave:

- Dado que <contexto inicial proposto>
- Quando <evento ocorrido>
- Então <resultado esperado>

5.4 Reuniões de elicitação

"Se você tem mais de 10% de requisitos mudando à medida que avança, você os especificou muito cedo. Se você tem ciclos de testes e correção separados você está testando tarde demais."
– Mary Poppendieck

Há uma infinidade de técnicas ágeis de elicitação. Algumas delas eu detalho nos próximos capítulos, outras citarei aqui e, conforme o caso, cabe a você dar uma pesquisada. Sugiro participar mais de grupos de usuários e de prática, comunidades de conhecimento e fóruns. Chamamos assim aos eventos necessários à fase de Discovery, para alinhamento, elicitação, debate e necessidades, sonhos, superpoderes entre os diferentes envolvidos no produto ou projeto. Por exemplo:

- **Entrevistas** – É consenso que dinâmicas em que mais de um envolvido estejam presente são mais produtivas, diminuindo hipóteses pessoais e ruído na comunicação;
- **User Story Mapping** – Uma documentação assertiva pela visão do usuário;
- **Brainstorming** – Reunião para debater soluções ou oportunidades; há variadas técnicas para organização e registro, como Managing Dojo do Manoel Pimentel ou Learning 3.0 do Alexandre Magno;

- **Focus Group** – Técnica de apoio à decisão, o objetivo é apresentar sua proposta ou solução para que os usuários validem e completem seus pressupostos;
- **Grooming** – Reuniões com participação do time para refino da documentação que está sendo construída no Discovery junto aos usuários. Não é uma timeboxe, mas mesmo não constando como parte do framework, há recomendações de que estas reuniões de Grooming poderiam ocupar até 5 ou mesmo 10% do tempo de um time durante um Sprint.

5.5 Mínimo Produto Viável (MVP)

"Se você não pode descrever o que você está fazendo você não sabe o que está fazendo."
– W. Edwards Deming

MVP (do inglês *Minimum Viable Product*), Mínimo Produto Viável é uma estratégia usada para validar produtos ou funcionalidades com o seu mercado, popularizada por Eric Ries. Trata-se de uma técnica iterativo-incremental de verificar o máximo de pressupostos a cada ciclo, revisando e adaptando a estratégia de acordo com as informações colhidas e privilegiadas durante cada ciclo deste processo.

Em um mercado globalizado, altamente competitivo, com oportunidades que mudam cada vez mais rápido, ter uma ideia e trabalhar durante meses (ou anos) para lançá-la é uma estratégia que potencializa riscos, não importa se falamos de uma startup ou multinacional.

É mais fácil ter ideias grandiosas do que pequenas e focadas, diretas ao ponto, e saber extrair entre 50 funcionalidades o que é de mais valor, aquele pedaço do todo que torne viável confirmar se a ideia será um produto de sucesso. Mais que isto, é **resistir** à tentação de "já que" está fazendo o MVP, perder a noção de tempo e acabar fazendo estoque.

Eu gosto de usar como exemplo aquela startup de nerds, que têm uma ideia, têm qualidade de código, testes unitários, produtividade, conseguem

500 mil de investimento e iniciam uma hibernação dentro de uma sala, trabalhando meses em um produto que "acham" que vai bombar, que será uma revolução, mas ao lançar descobrem tarde demais que não rolou.

A especificação e construção de um MVP é a busca por uma versão que seja o menor possível, mas que contenha uma proposição de valor relevante para o cliente, potencializando venda e fidelização devido a funcionalidades e características. São mínimas, mas que apontem claramente para um produto consistente e com potencial de alto valor agregado.

Atenção

MVP não é a simples construção e entrega do mínimo possível, construído no menor prazo, menor custo ou com as funcionalidades de menor complexidade. A equação correta é a busca de equilíbrio entre um produto de real valor para o cliente, construído em tempo e com recursos limitados, mas que confirme sua viabilidade como negócio. Na mesma equação está o uso de ferramentas open-source ou serviços free, com agilidade e ciclos iterativo-incrementais que antecipem pressupostos acerca do produto concebido antes de desenvolvê-lo.

5.6 User Story Mapping

[quote "Não é o suficiente fazer o seu melhor, você deve saber o que fazer, e então fazer o seu melhor." W. Edwards Deming]

Com certa frequência alguém me pergunta sobre conhecer ou não uma técnica indicada para o levantamento de requisitos em uma cultura ágil. A primeira que me vem à mente é a *User Story Mapping*, criada por Jeff Patton.

Elicitação x US Mapping

No antigo mindset, o início se dava o processo de elicitação, entrevistas individuais e reuniões com stackholders. Depois, o analista, em sua sala ou baia, passava dias transcrevendo, diagramando, colocando no papel seu entendimento e, conforme a urgência, saía com o cronograma com tarefas.

A US Mapping é uma técnica ágil utilizada para levantar e organizar requisitos, que exige apenas a combinação de um mediador experiente, uma parede (tão grande quanto o projeto), muitos post-its, canetões e fita crepe. Os participantes são os convidados que podem e devem contribuir para a visão dos desejos relativos ao produto.

Quanto ao número de pessoas, já participei de US Mapping com mais de dez pessoas, entretanto, creio que se ultrapassar muito disto há o risco de tornar-se muito lento ou confuso. Achei referências de até 12 pessoas como limite, mas há casos de reuniões desta natureza com mais de 12 que funcionaram.

Quanto ao tempo, pode durar um turno, um dia ou mais, conforme estiver claro o entendimento do que esperam do produto. Depende de quão complexo é e da colaboração das pessoas certas – PO, UX, SEO, executivo, *keyusers*.

Preparação e ambientação

Use uma mesa em que todos estejam bem confortáveis, junto a uma parede bem grande e desimpedida, sem móveis ou quadros. Dê um bloco de post-its amarelos para cada um, um canetão com ponta fina, e alinhe com toda a equipe como as US devem ser escritas.

Na parede, cole uma fita crepe montando um eixo cartesiano (xy), sendo o x um prisma pela **sequência** de eventos (por exemplo, um site de busca tem a necessidade de uma home com a caixa de busca antes da página de resultados), e o y um prisma de **relevância** ou **valor**:
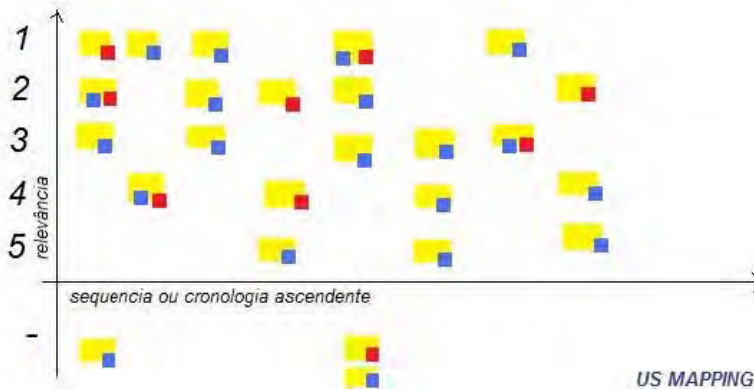
Fig. 5.2: User Story Mapping

Inicie com uma apresentação geral do que já se sabe para entendimento da estratégia, pontos de atenção, oportunidades e riscos. É bom manter um clima aberto, sem pressão ou repressão. Todos podem sugerir e falar livremente, desde que haja bom senso, educação e convicção.

Desenvolvimento

Defina uma ordem em que cada um tenha sua vez de sugerir uma US, de forma cadenciada e sem estresse. A comunicação e entendimento de cada US é importante para evitar esquecimentos ou redundância. Forme uma fila circular em que todos sugiram uma nova US ou passem a vez para o próximo, caso não tenham nada para sugerir.

A cada post-it (US) sugerido, o próprio integrante que a propôs ou o mediador a coloca em uma posição dos eixos xy e rapidamente todos podem colaborar para sua disposição, mais para cima ou para baixo, caso seja mais ou menos relevante, e direita ou esquerda, para refletir sequencialidade. A cada modificação, é possível que não só este post-it seja reposicionado nos eixos xy, mas que outros tenham que se adaptar comparativamente. Isso é mais que previsível, posto que é impossível coincidir que os post-its (US) surgirão do mais relevante para o menos e do primeiro ao último na sequência.

Informações adicionais podem ser combinadas e introduzidas através de post-its pequenos de outras cores, destacando situações especiais importantes

e já percebidas em alguma US, como o tamanho PMG, valor BMA, responsável etc. Todas as informações e ajustes das posições podem mudar, pois à medida que novos post-its (US) entram, a comparação entre eles fica cada vez mais clara.

Concluindo

Conforme evolui a reunião, o quadro vai se compondo e formando uma linguagem ubíqua. O argumento ou debate realizado a cada rodada gera sensação de unidade e convergência entre participantes, bem como reforça a pertença e clareza nos porquês de valor.

É importante salientar que, no caso de post-its (US) que tiveram baixa priorização ou foram desconsiderados, abaixo do eixo x, tendo recebido debate apropriado, na falta de argumentação suficiente, ficará clara a todos, inclusive ao seu "criador", a fragilidade de seus argumentos, cabendo melhorá-los ou desqualificá-los.

Por outro lado, o fato de não estar na primeira linha ou na primeira coluna não quer dizer que ficará para depois ou não será feito. A montagem de Releases e Sprints possui variações, como requisitos, dependências, oportunidades, tecnologia, oportunidades e riscos.

Planejamento de Release

Por fim, o planejamento de Releases, e talvez já esboçando seus Sprints, é feito em um esforço conjunto que busca dividir o todo em pacotes, explicitando as prioridades e correlações importantes. Desenha-se uma linha de tempo para suas entregas, com uma ou mais Releases, cada uma com uma ou mais Sprints.

O processo para definição de Sprints começa no primeiro post-it à esquerda no topo. A partir dele, faz-se uma leitura radial, lembrando que se trata de uma negociação. Os mais altos são mais valorosos e os mais à esquerda são requeridos antes. O objetivo desta última fase do User Story Mapping é visualizar quantas são e qual o escopo inicial de cada um dos Sprints necessários.

Lembre-se que foram convidadas as pessoas certas, conhecedores do negócio, experientes na tecnologia. Eles se empenharão em distribuir da forma

mais racional e montar um bom plano de construção iterativo-incremental, para que entreguem o melhor e máximo valor a cada ciclo, de uma maneira possível e sustentável.

5.7 Product Backlog e Sprint Backlog

"Genba significa que devemos estar onde as coisas realmente acontecem, envolver-se pessoalmente, na hora e local apropriados."
– Taiichi Ohno

A partir de reuniões ou técnicas ágeis de levantamento de requisitos da solução desejada, deve ser montada uma lista de demandas devidamente organizadas e priorizadas, um instrumento vivo, que deve ser revisitado e ajustado à medida que pressupostos do produto ou projeto são validados.

O responsável pelo Product backlog é o Product Owner e as ferramentas utilizadas para repositório são as mais variadas, desde o MS-Excel, mapas mentais, soluções especialistas que suportam projetos ágeis de desenvolvimento ou mesmo a opção pelo uso de uma parede e muitos post-its.

A construção do Product backlog a partir de técnicas como a User Story Mapping e outras dinâmicas de elicitação ou brainstorming deve gerar listas com um consenso de prioridade, valor e complexidade, à luz da visão e expectativa dos principais envolvidos.

Enquanto o Product backlog contém a totalidade de requisitos e desejos já expressos pelos envolvidos, o Sprint backlog é o *subset* pretendido para o próximo Sprint e que será confirmado ou não durante o pacto construído no planejamento do Sprint.

Capítulo 6

Delivery

Ao saber o que e por que construir, a equipe estará focada em entregar aquilo que foi pactuado para este momento, com seu melhor design e qualidade técnica-funcional. O time trabalhará sob um prisma colaborativo, antecipando oportunidades e mitigando diariamente riscos percebidos, por meio de um método que prima pela transparência e realismo como meio de agregar valor antecipadamente e eliminar todo e qualquer desperdício.

6.1 Sprint Planning

"O conhecimento era um bem privado, associado ao verbo SABER. Agora, é um bem público ligado ao verbo FAZER."
– Peter Drucker

A experiência mostra que uma boa reunião de Planning é para ser produtiva, eficiente, mas não precisa ser tensa ou chata. O melhor turno é o da manhã, com todos tranquilos. Com frequência organizamos um café antes, entre aqueles que querem chegar um pouco mais cedo e descontrair.

1) Iniciar alinhando datas, o que já foi e o que falta;

2) Cálculo de velocidade e DoD (*Definition of Done*);

3) Apresentação das User stories e estimativas;

4) Transcrever e subtotalizar tudo no quadro;

5) Manter análise da ocupação e extras, P&D, Pair etc.;

6) Pacto de trabalho para o sucesso do Sprint.

Durante a apresentação feita pelo PO, a participação do UX, SEO, arquiteto ou quem mais tenha participado decisivamente do detalhamento de cada história é fundamental, pois o pleno entendimento de cada detalhe e variantes do layout e arte desenvolvidos é crítico para que as estimativas sejam mais realistas possíveis.

São 4 horas: 2 para a apresentação do **que** precisa ser feito e qual o valor, seguidos de até mais 2 horas no debate para estimativas do **como** será feito. Entretanto nós usamos uma variação, pois as histórias são apresentadas em pequenos grupos relacionados, seguidos de debate e estimativa, evitando *delay* com repetição de perguntas ou esquecimento de respostas.

Na prática, se após algum tempo de Planning sentimos que existem mais dúvidas que entendimento, que o Discovery foi muito frágil e que não há como estimar as histórias, transforma-se aquela reunião em uma reunião de Discovery. Fechamos as pontas e fazemos o Planning na sequência ou na manhã do dia seguinte, o que evita frustrações, estresse e riscos.

Se as dúvidas forem poucas, é possível dar algum tempo para a equipe entender bem o que se quer ou mesmo debater com o PO para definir alterações para tornar aquela história factível ou mesmo deixar em aberto e realizar uma continuação do Planning no dia seguinte, desde que o tempo máximo não se estenda demais. É uma questão de bom senso.

Fundamental: o Sprint Planning é visual

Se não houver uma parede forrada de fórmica ou quadro branco, use post-its e folhas de *flipchart* colados na parede, pois é lá que transcrevemos tudo o que é debatido, decidido e todos os números. De forma visual, constrói-se uma única visão do todo e suas partes, destacando gargalos e oportunidades, um pacto.

Quem olha para o quadro, verá nele absolutamente tudo o que foi tratado, com seus detalhes. É útil quando voltamos a algum item já discutido ou quando traçamos uma linha relacionando diferentes itens ou observações. Às vezes colamos o desenho de interface, com as telas ou blocos que construiremos; assim tentamos que nada passe despercebido e que o sucesso de cada parte envolva as demais.

Sugestão final: tenham sempre uma câmera de qualidade à mão. Não importa o tamanho, mas sempre fotografe suas reuniões, cada parte do quadro. Coloque em um repositório, deixe acessível a todos. Poderá ser utilizado para dirimir alguma dúvida, posto que tudo o que é acordado ou debatido está lá representado.

6.2 Planning Poker

"Não seremos limitados pela informação que temos. Seremos limitados por nossa habilidade de processar essa informação."
– Peter Drucker

Há uma infinidade de oportunidades no uso de escalas de complexidade para dimensionamento e monitoramento de estimativas, construção e entregas. Não é uma técnica do Scrum, é uma alternativa para estimativas em um time ágil. Exige algum tempo extra, mas é divertida e possui ganhos na atenção aos princípios e papel de cada integrante nesta importante tarefa inicial do ciclo de Delivery.

O *planning poker* usa o conceito de complexidade. Para exercitar corretamente este conceito, existem baralhos com um subset semelhante à série de Fibonacci = 1, 2, 3, 5, 8, 13, 21, sempre somando os dois últimos números para chegar ao seguinte. A sequência refere-se a uma provável curva crescente de

incerteza, quanto maior uma tarefa, maior a incerteza, aumentando significativamente seu tamanho em pontos. A série acaba impondo esta curva, por exemplo, maior que 3 é 5, acima deste já é 8, depois 13.

É uma dinâmica coletiva, onde todos podem e devem se posicionar após bom entendimento do que é cada funcionalidade ou User story. Ao estimar, é necessário comparar com as estimativas anteriores, mantendo-se a coerência e refazendo quando necessário. Para começar, sugere-se que a equipe escolha a menor de todas as histórias, usualmente um pequeno cadastro, atribuindo-se 1 a ela, de forma a servir de baliza para este início de estimativa.

Nos baralhos distribuídos por grandes empresas de software e treinamentos, há sempre cartas com um ? para quando não nos sentimos em condições de estimar, uma de "café" para pedir uma pausa e em alguns casos cartas acima de 21 pontos. No caso de User stories, devemos sempre nos questionar se não podem ser quebradas em histórias menores, posto que um dos princípios para estimativas ágeis é termos um nível de granularidade adequado, por isso chamamos de User stories. É para cada uma relatar apenas uma interação do usuário. As regras para o planning poker são:

- Cada integrante da equipe de desenvolvimento recebe um baralho de Planning Poker;
- É importante haver uma combinação do número a partir do qual devemos tentar quebrar em mais histórias;
- A equipe seleciona a menor história a ser estimada como 1 para servir de referência no início;
- A partir desta história mensurada como 1, vamos estimando as maiores, comparativamente;
- Escolhe-se uma história, pergunta-se se a galera entendeu e pede-se para que escolham uma carta;
- Quando todos escolheram a sua carta, pede-se que mostrem para os demais;
- Se houver consenso, está estimado e vamos adiante;

- Se não houver consenso, qualquer um, mas em especial os extremos devem justificar sua opção;
- Após os argumentos, roda-se mais uma vez;
- O limite hipotético é de 3 rodadas, quando então a estimativa segue a carta mais alta.

É uma técnica que valoriza o senso de pertencimento, aprendizado contínuo, multidisciplinaridade e consenso. Ela destaca eventuais más interpretações e conflitos de percepção através do debate entre profissionais com diferentes expertises. Vale a pena tentar, mas não é mágico, nas primeiras vezes podemos errar, mas fiquem tranquilos, com o tempo erraremos menos. Se passarmos a acertar sempre, desconfie, porque estimar software é uma tarefa complexa, mais provável ser um sinal que estamos com uma boa e generosa margem.

6.3 TDD: Test Driven Development

"Insanidade: é fazer a mesma coisa dia após dia e esperar resultados diferentes."
– Albert Einstein

Kent Beck desafiou a todos em 2003 com uma abordagem onde qualidade não pode ser avaliada ou pretensamente alcançada através de testes em um produto já construído.

Com o uso de TDD temos por objetivo prevenir defeitos em primeiro lugar, desde antes e durante toda a construção do software, permitindo uma medida real de qualidade, que pode ser assim traduzida:

1) A cada decisão construída e não testada, existe uma grande probabilidade de a decisão ou de sua implementação estar errada;

2) Funcionalidades de software que não podem ser demonstradas através de testes automatizados não são confiáveis;

3) Testes garantem que refletiremos sobre o que será e aonde queremos chegar, independente da forma como a solução será implementada.

Comecei a entender o que de fato era TDD após assistir um Coding Dojo. Você inicia pela classe de teste, evoluindo seu código aos poucos, sempre a partir do seu teste unitário. Algumas vantagens desta abordagem são:

- Ajuda na documentação: testes bem construídos são mais simples de ler pela equipe do que o próprio código, como se fossem critérios de aceitação;
- Incentiva a simplicidade: como a solução vai surgindo pouco a pouco, a tendência é que se foque no imprescindível;
- Aumenta a confiança: todos os testes são montados antes e durante a construção do software, dando-lhe maior confiança;
- Facilita *refactorings*: quanto mais testes existem no sistema, maior é a segurança para fazer e entregar refactorings.

O código de testes tem duas funções principais: de especificação, para definir uma regra que seu software deve obedecer; e de validação, para verificar que a regra é obedecida pelo software.

Para tornar a construção de testes mais produtiva, existem no mercado frameworks e plugins, como por exemplo os Mocks, objetos que simulam o comportamento de objetos reais de forma controlada.

O processo de criação de programação orientada a testes segue um ciclo contínuo de desenvolvimento, conforme a seguir referenciado. Estes três passos são repetidos até que não se consiga pensar em novos testes, pois a funcionalidade estará 100% desenvolvida e testada:

1) Escrevemos um teste que falhe, ainda para uma classe/método que não existe. Como se fôssemos declarar os critérios de aceitação, pensamos primeiro no teste e, só depois que ele estiver pronto, criamos somente o suficiente de código necessário para que ele compile e falhe ao rodar.

2) O passo seguinte é fazer o teste passar: construímos um código de forma que o teste rode e passe.

3) Agora que temos um código funcionando, refatoramos e o melhoramos um pouco, otimizando-o naquilo que for possível, em pequenos passos.

Há desenvolvedores que passam meses, anos, planejando iniciar a usar TDD, testes unitários, escolhendo um mock, mas, se você quer saber, o primeiro passo são Coding Dojos. É como jogar pingue-pongue sem nunca ter pegado em uma raquete. Comece treinando de forma descontraída, com amigos, sem pressão ou cobrança, vendo quem sabe e tentando segui-los.

Para começar a treinar com dojos, é preciso desmistificá-los. Há um site só de desafios sugeridos, onde se diz que as sugestões foram utilizadas em 1577 dojos. Eu acreditei: http://dojopuzzles.com. Veremos mais sobre Dojos na seção 7.4.

6.4 Scrum board (quadro de tarefas)

"Se você não sabe para onde você quer ir, qualquer caminho você pode seguir e nenhum mapa irá lhe ajudar."
– Lewis Carroll, Alice no país das maravilhas

O quadro de tarefas, conhecido também como Kanban, contém colunas de evolução de status e papéis coloridos representando tarefas. Elas iniciam em um status de **a fazer**, evoluem para **em construção**, para depois serem liberados e seguirem para outros status como **testes**, **validação** e, finalmente, **entrega**.

Variações

Para cada equipe, conforme o tipo de negócio, produto, maturidade, ambiente, entre outros fatores, teremos variações de quadros, de 3 colunas com *to do*, *doing*, *done*, até aqueles com mais de 10 colunas e quadros adicionais para impedimentos e outros.

Desenvolvimento			Testes		Homologação		Produção		
To Do	Doing	Wait	To Do	Doing	To Do	Doing	DoD	GMUD	Deploy

Fig. 6.1: Quadro Kanban

Forma

Não recomendo a ninguém adquirir ou adotar apenas quadros virtuais antes de reter alguma experiência e maturidade com papeis coloridos, flags, etapas, avatares, para compor uma identidade própria e única. Em meio ao seu uso, cada detalhe, cor, ponta torcida do papel, sobreposição de selos (bug, mudanças, combinações etc.) tem especial clareza e sentido.

Papel

A opção mais racional que encontramos é imprimir em folha A4 colorida o layout exato de nossas representações de tarefas, com lugar para as informações necessárias – projeto, iteração, número, título, estimado, prioridade, complexidade etc. Após impresso, guilhotinamos 8 folhinhas, escrevemos nelas com canetão e colamos com fita crepe.

Organização

Lembrem-se de uma premissa importante dos métodos ágeis, os instrumentos usados pela equipe devem ter valor percebido pela equipe e cabe à equipe interpretar a melhor forma de uso. A imposição de modelos externos tendem a ter maior resistência do que os auto-organizados.

Três colunas é pouco

Um bom quadro Kanban, que explicite as etapas de construção e seja enriquecido com avatares, post-its formatados, selos, marcações, limites, WIP (*work in progress*) etc., nos leva a um Sprint claro e transparente, otimizando a comunicação necessária antecipação de riscos e oportunidades.

Se feito corretamente, o quadro torna possível a qualquer integrante ou interessado ter uma clara informação do Sprint e seu andamento apenas passando os olhos por ele na parede. Bons quadros e gráficos consomem algum tempo de start, configuração, montagem inicial, sim, mas após isto são poucos minutos diários para mantê-los atualizados.

Até onde ao adotar um método ágil o fazemos no mínimo exequível e buscamos novas zonas de conforto? Será que estamos evitando experimentar coisas diferentes, quer sejam artefatos, técnicas ou atitudes? Não existe mudança de metodologia sem esforço adicional para mudança de hábitos, sem desapegar do velho para aprender o novo.

6.5 Tarefas

"O principal objetivo do Scrum é ajudar as equipes a concentrarem-se em seus objetivos."
– Ken Schwaber e Jeff Shuterland

O uso de post-its coloridos adquiridos em livrarias pode ser um desperdício de oportunidade. Logo após a adoção do Scrum, eu desenhei um modelo de "post-it" impresso em folhas A4 coloridas, com oito por folha e que após impresso são guilhotinados.

O custo da guilhotina retorna em dois ou três meses, pois os packs de folhas A4 coloridas são muito mais baratos que pacotes de post-its, com o ganho de podermos personalizá-los de acordo com o layout acordado com a equipe de forma a tirarmos o máximo deles.

A seguir, um exemplo de "post-it" já guilhotinado em uso, onde se percebe que usamos uma linha para cada dia útil do Sprint, de forma que ficam muito claros quais os dias alocados e quantas horas em cada dia. Outra sutileza é que podemos ter duas pessoas apropriando, seja em *pair*, ou um backend e outro frontend, evitando redundância de post-its no quadro:

Fig. 6.2: Pré-impressos para tarefas

Usamos "post-its" de duas cores: azuis para tudo o que foi planejado no Sprint planning e amarelo para todo e qualquer extra ou tarefa imprevista realizada durante o Sprint, algo comum por sermos uma equipe que realiza projetos e faz a manutenção do produto atual em produção.

Outra oportunidade é o uso de selos coloridos, para os quais também usamos folhas A4, como um inseto para bugs, um número de 1 a 9 para prioridades, avatares com personagens (impressos e coloridos), selos de impedimento com data de início, responsável, liberação etc. Isso varia de time para time, de acordo com os acordos feitos nas retrospectivas.

6.6 Daily

"Nem tudo que se enfrenta pode ser modificado, mas nada pode ser modificado até que seja enfrentado."
– Albert Einstein

É uma reunião diária de no máximo 15 minutos, sempre que possível executada no mesmo horário, todos os dias, com todos os integrantes da equipe em frente ao seu quadro de tarefas.

Quando alguém estiver ausente, em especial o Product Owner, tentem colocá-lo em viva voz (mobile, skype etc.), de forma a compartilhar o pro-

gresso, oportunidades, dificuldades de cada integrante e a definição de microplanos de ação, visando o atingimento da meta do Sprint.

Pode-se tomar a *daily* como o **termômetro** do método: se houver 2 ou 3 destas reuniões sem a necessidade de tomadas de decisões, revisão da estratégia, antecipações ou impedimentos, tenham a certeza de que algo está errado, os três pilares do SCRUM – transparência, inspeção e adaptação – não estão sendo praticados, falta entendimento ou crença.

Se as reuniões estão sendo efetivas, garanto que não existirão dois dias iguais, cada daily é única. Uma *Daily Stand Up Meeting* de 15 minutos possui 3 momentos distintos, um tanto quanto óbvios, mas que precisam ser entendidos e realizados da melhor forma possível:

1) **Antes da daily** – Ninguém deveria ir para a reunião sem ter refletido antes no que vai dizer. Oriente para que as pessoas insiram um alerta no "Outlook" para lembrar de se prepararem, a pior coisa que tem é alguém dizer "Não lembro o que fiz ontem!" ou "não sei como está meu impedimento!";

2) **Durante a daily** – Devemos ser o mais transparente e engajado possível, ter realmente interesse no que as pessoas estão dizendo, pois a contribuição de cada um influenciará no atingimento da meta da equipe. As informações devem ser assertivas, diretas, claras, o foco principal é identificar gargalos e garantir o sucesso do Sprint;

3) **Depois da daily** – Com certa frequência, identificamos na daily a necessidade de um ou mais debates, detalhamento, revisão de análise ou mesmo estratégia, que devem ser assinalados para serem realizados após a daily. Pode ser ali mesmo, em frente ao quadro de tarefas, em uma mesa ou mesmo em uma sala.

A seguir algumas dicas para a execução da daily:

- **Local** – Deve ser o próprio ambiente de trabalho ou, no caso de reuniões remotas, uma sala de reuniões próxima, para uso de viva-voz;
- **Quórum** – Todos os integrantes da equipe devem participar, todos os dias, presencialmente ou remotamente (se inevitável);

- **Ordem** – Em uma equipe de produtos digitais para web eu sugiro iniciar pela equipe técnica, SEO, UX, PO e, por último, o Scrum Master;
- **Token** – É interessante e divertido definirmos um objeto relacionado ao negócio ou produto, que identifique com quem está a palavra, ninguém mais;
- **Assertividade** – É importante assinalar claramente ou fazer coaching com os mais prolixos, dispersos ou descrentes da utilidade das timeboxes;
- **Quadro** – A daily e o quadro de tarefas se completam como queijo e goiabada; visam facilitar a explicação e o entendimento, vitais à transparência;
- **Clima** – Devemos trabalhar para que não seja uma reunião tensa, de cobrança ou aflita, pois deve ser construtivista e em prol da auto-organização.

6.7 Burndown

"A melhor maneira de começar a implementar o Scrum é estabelecer reuniões diárias de status."
– Ken Schwaber e Jeff Shuterland

O gráfico diário de *burndown* não é um artefato opcional, é uma peça-chave na leitura e entendimento do quadro de tarefas e vice-versa, um indicador de tendência para atingimento dos objetivos pactuados no Planning.

O quadro de tarefas não indica a tendência entre previsão e trabalho. A única forma de ter esta informação sem o burndown é somando os post-its e analisando-os sob diferentes aspectos, mas com o burndown atualizado, esta informação estará sempre acessível.

Iniciamos o burndown ainda no Planning, com o cálculo de velocidade, listando os integrantes, a disponibilidade, ausências, tempo útil e livre. É importante desconsiderar os tempos necessários para o Planning, a Review, a Eetrospectiva, alguma pendência em curso e eventuais compromissos relacionados ao *OnGoing*.

A taxa de horas úteis é particular a cada time e muda conforme passam os Sprints. Uma equipe pode ter 8:30 brutas diárias, 7:00 úteis e 1:30 livres, destinadas a questões administrativas, pequenas reuniões, atendimentos, apoio a colegas.

Vamos assim montando cada burndown (eu uso um para cada especialidade) com um risco contínuo para a velocidade (horas úteis) e um tracejado bold para as horas estimadas na construção das US e tarefas discutidas, oportunizando antever e debater oportunidades (como pair, P&D) ou riscos (superalocação e desequilíbrios).
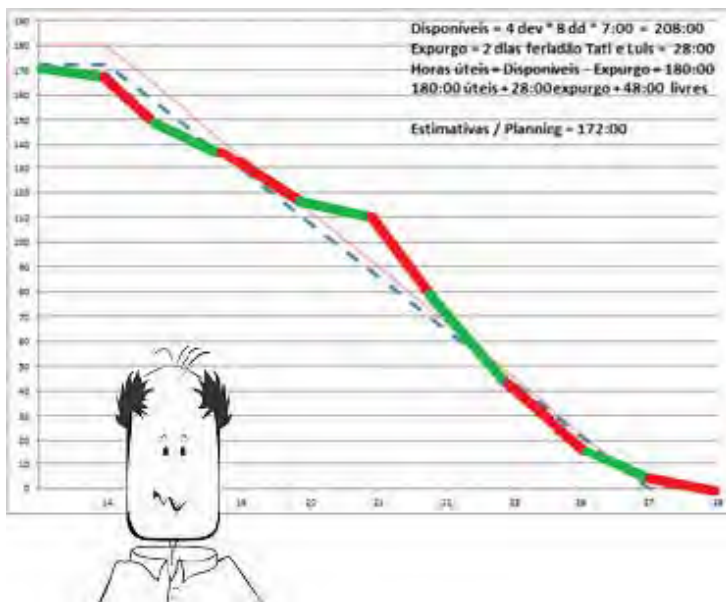
Fig. 6.3: Diagrama de Burndown

Devem-se somar todas as estimativas ou reestimativas (coluna "falta" dos post-its) e marcar no burndown, mas atenção: o burndown só utiliza o tempo faltante, aquilo que falta fazer, como gráfico de "tendência". O objetivo é ver se tudo o que falta cabe no tempo disponível ou instigar a negociação e planos de ação.

No exemplo, no 1º e 2º dias estávamos bem, mas o 3º e 4º apontavam uma tendência a estourarmos o tempo; no 5º dia, a tendência se confirmou e estourou, mas a reversão no 6º dia indicava um plano de ação para resolver ou mitigar o problema, trazendo a tendência para uma perspectiva de sucesso. Transcorremos ok até o penúltimo dia, quando algum problema impactou, mas que não comprometeu a entrega final.

Riscos e oportunidades

Construir um burndown por história só faz sentido se suas histórias são todas pequenas e de uma granularidade homogênea. Construí-lo por tarefa é um pouco melhor, seguindo a mesma premissa, com tarefas pequenas e homogêneas. Se vocês tem o hábito de estimar em horas, o burndown é tão preciso quanto a técnica e artefatos permitem.

A granularidade e unidades de estimativa geram burndowns meio inúteis como as imagens mais a esquerda, ou o da direita, que nos dá uma tendência real e precisa a cada dia do nosso Sprint. A maior parte das equipes possuem dificuldades em quebrar histórias em tamanho adequado, ou na quebra de tarefas. Outro empecilho são softwares utilizados para gerar burndowns úteis ou inúteis.

Multidisciplinaridade

Outra maluquice que na teoria se sustenta, e na prática não, é gerar um único burndown para a equipe, somando os tempos ou unidades sem diferenciar alocações de desenvolvimento, testes, analistas, como se todos já fossem multidisciplinares. Esse hábito pode ocultar tempo estourado de desenvolvimento e de sobra em testes, ou causar desequilíbrios piores ainda caso tenha outras divisões.

Em 30 anos em desenvolvimento de software não tive experiências de plena multidisciplinaridade, pelo contrário, analistas, desenvolvedores e testadores ainda são uma realidade por opção técnica da maioria das empresas. Fechar os olhos para isso e somar tudo em uma única linha de burndown pode induzir a achar que está tudo bem enquanto não está faz tempo.

Finalmente, minha abordagem é simples demais: ter um burndown não conclusivo é um grande desperdício. Ele só tem valor real se qualquer inte-

grante puder "lê-lo" e tirar todas as conclusões por si. Se precisa de um manual e alguém experiente que cruze dados e chegue a conclusões baseados em sua experiência é sinal inequívoco de que está tudo muito errado.

Não existe quadro Kanban que nos mostre facilmente o que o burndown oferece. No quadro é possível ver gargalos, ocorrências, fluxo, mas ele não pode nos dar tendência de sucesso cruzando o quanto falta *versus* o tempo que ainda temos até o final do Sprint.

Eu acredito em estimativas como uma fonte de autoconhecimento, até concordo que empresas que possuam equipes de alta senioridade e muita experiência em metodologias ágeis possam abdicar de estimar e basear-se apenas no sentimento do quanto conseguem fazer e fazem, mas decididamente não é o meu caso.

Não sou xiita, se a cultura da empresa e seus protagonistas preferem trabalhar só com pontos ou nem isso, tudo bem. Mas não é o que recomendo e não reflete o sucesso que tive em implantações difíceis com equipes que estavam adotando Scrum e iniciando uma trajetória ágil em empresas que necessitavam de métricas inteligíveis.

6.8 Artefatos adicionais

"Obstáculo é aquilo que você enxerga quando tira os olhos do seu objetivo."
– Justin Herald

Um dos fundamentos dos métodos ágeis é a cadência, o fluxo contínuo que deve ser trabalhado desde a fase de *Discovery* até a de *Delivery*. Mas quero destacar o trabalho da equipe de desenvolvimento. É fundamental que tenhamos entregas diárias de User stories para testes e homologação, evitando gargalos ou estoques de partes inacabadas.

Um artefato importado dos cursos de Kanban é o gráfico de dispersão de *User Story Tracking*, que informa a quantidade de User stories ou tarefas entregues a cada dia. O objetivo desta visão é termos uma visão de quantas histórias em média estamos entregando a cada dia e a cadência atingida, dados que serão úteis em retrospectivas com o objetivo de entender melhor nossa velocidade.

Outro muito simples e fácil de manter é o de *Lead Time* ou *Cycle Time*, pois trata-se de uma grade com eixo x com os dias do Sprint e o eixo y com as histórias devidamente numeradas. A cada dia, vamos marcando com um ponto o início do desenvolvimento daquelas histórias que os desenvolvedores pegam. A cada dia não concluída, ela vai mantendo uma linha contínua, até ser concluída, recebendo um X no diagrama, equivalente a fim.

Em 2012, criamos uma *Daily Tracking* para manter registro do que está pegando nas dailys, algo que até facilitou a visibilidade das principais atividades de cada um, tempos apropriados por dia e exceções do dia a dia, ausências, férias, imprevistos alheios em tarefas previstas.

Algumas equipes integraram este *tracking* à sua daily em diferentes formatos. Trata-se de uma memória de cálculo; há times que optam por uma *Daily Tracking* individual, em que cada integrante passam a ter uma folha com a grade de dias do Sprint sob seu teclado, anotando nela as informações cotidianas que facilitariam a apropriação e não esquecimento de detalhes e informações úteis na daily e na próxima retrospectiva.

Capítulo 7

Melhoria contínua

Um dos principais valores da abordagem ágil para desenvolvimento de software é a geração de oportunidades e ambiente com foco em melhoria contínua, privilegiando sempre a interação entre as pessoas, tanto internamente ao time, quanto do time com seus clientes, fornecedores, parceiros e demais envolvidos. O objetivo é manter todos os interessados e envolvidos a par do que está para acontecer e acontecendo, eliminando surpresas e pressupostos em favor de compartilhamento de fatos e oportunidades.

7.1 Review

"Tudo que se pede é uma chance de trabalhar com orgulho."
– W. Edwards Deming

Esta é a timebox que garante o estreitamento de relações entre o time e seus usuários, evitando que a equipe só "conheça" o negócio através de filtros. O resultado deste erro estratégico todos conhecem, pois não podemos esperar entregar valor, pertencimento e inovação de quem apenas ouve falar do negócio e das necessidades através de terceiros.

É uma timebox inicialmente desprestigiada. Inconscientemente alguns papéis sentem-se desconfortáveis com a perda de representatividade ou riscos à sua zona de conforto, quando deixam de ser o único a falar em nome do negócio ou têm que se envolver mais que o mínimo necessário.

As justificativas são sempre as mesmas: "Não tenho tempo para mais uma reunião! Pode tocar adiante, confio em você! Vai lá e depois nos informa! Não posso, estou cheio de trabalho! Você já sabe tudo o necessário!"

Dinâmica

Seguindo uma premissa importante de qualquer timebox, a condução deve ter em vista a natureza de sua pauta, deve ficar na apresentação do previsto e do realizado. Não é hora para fazer testes e homologação, nem para revisar ou discutir estratégia ou filosofar sobre o Product backlog.

Uma reunião de Sprint Review põe na mesma sala todo o time Scrum e todos os seus principais usuários, com uma pauta exigente a ser cumprida, pois primeiro o PO apresenta os objetivos e depois a própria equipe de desenvolvimento apresenta o que acabou de ser concluído no Sprint.

- **Usuário** – Reunir-se com todo o time, posicionar-se, perguntar e responder direto na fonte gera um vínculo importante e forma um espiral de autoconhecimento e responsabilidade, de compromisso e pertencimento.

- **Product Owner** – Permite um maior alinhamento da equipe ao negócio, valores e princípios, que facilitará muito a sua apresentação de futuras User stories e construirá um ambiente muito mais colaborativo em torno do que realmente é valor a ser atingido.

- **Equipe** – Entenderá mais do negócio, integrando mais o ecossistema, facilitando a comunicação em todos os sentidos, agregando maior

compromisso pessoal com a entrega de cada Sprint, pois é a própria equipe que apresentará e dará eventuais explicações.

7.2 Entrega de pacotes

"Competência é tomar a iniciativa e assumir a responsabilidade diante das situações profissionais com as quais nos deparamos."
– Philippe Zarifian

Iniciado um novo Sprint, a cada User story construída, testada e homologada, devemos sempre avaliar entregas frequentes com o negócio e equipe, diárias, podendo inclusive ser mais de uma ao dia, se possível, evitando fazer estoque de solução, na meta de diariamente entregar valor.

A decisão está atrelada a questões técnicas, estratégicas e de negócio, oportunidades e mitigação de riscos. Podemos reter a publicação em função de quotas comerciais, lançamento, mas no outro extremo poderemos colocar em produção cada pequena melhoria possível. Cada caso é um caso.

Os princípios por trás do método Scrum garantem a comunicação e cadência no cotidiano de uma equipe de desenvolvimento em seu ecossistema. Esta é a natureza de cada timebox; não é preciso aguardar a Daily para tomar uma decisão, a Review para fazer uma entrega, a Retrospectiva para uma melhoria.

No tocante à entrega de pacotes, o Norte é qualidade e valor entregue. Empresas de todos os portes já utilizam conceitos de testes automatizados e *continuous delivery*, desde players como Facebook, Yahoo, Microsoft até uma infinidade de StartUps entregam vários pacotes a cada dia.

Contudo, atenção! O pulo do gato não é o uso de softwares de integração como o Hudson ou Jenkins, que permitem uma integração contínua, mas um robusto modelo de testes automatizados. Em pleno século XXI, a maioria dos profissionais ainda não adotou testes unitários e de regressão.

A tempo, a colocação automática em produção não é uma meta para todos. Conforme o negócio, o produto, a estratégia, talvez seja necessário um planejamento, um agendamento, mas isso não impede que o processo de publicação seja baseado em um processo automatizado.

7.3 Retrospectiva

"A mente que se abre a uma nova ideia jamais voltará ao seu tamanho original."
– Albert Einstein

A reunião de retrospectiva é a nossa **bússola**. A cada Sprint, a equipe se reúne para ver de onde venho, onde está e para onde quer ir. Precisa avaliar como trabalhou e se atingiu os objetivos, como melhorar, como ser mais feliz fazendo o que faz e como tentar executar com mais qualidade e produtividade, pois valor para o negócio também faz parte de nossos sonhos.

Algumas premissas básicas: ser uma reunião construtiva, com foco no futuro e não lavagem de roupa suja ou busca de culpados. Também é dar espaço; não podemos forçar ninguém a falar. O tempo e a maturidade no método se encarregarão de achar a hora certa de a equipe se abrir e falar às claras sobre suas dificuldades, acertos e erros.

Exemplo de programação para uma Retrospectiva

- **Quebra-gelo**: use uma bola ou rolo de fio de mão em mão em ordem aleatória e peça para cada um falar alguma coisa de si, algo fora do usual, como "Hobbies?" ou algo da sua história profissional como "O que deixou passar e hoje não mais deixaria?" ou "O que topou e hoje não toparia?";

- **Valores**: coloque números de 1 a 5 no chão e pince algumas frases sobre os valores e crenças da empresa ou negócio, peça que para cada frase, cada um fique no índice de adesão, número 1 se não fazemos nada, até o número 5 para caso façamos muito o que a frase diz (Ex.: Resultado a qualquer custo!);

- **Processo**: análise do processo e artefatos gerados durante o último Sprint para identificar pontos de melhoria a partir daqui, com foco em valor, qualidade, produtividade e prazer em fazer. Devem-se registrar as sugestões e acordar alguns planos de ação, com data e responsável;

- **Fechamento**: uma avaliação da reunião de retrospectiva que está acabando, insights, perspectivas percebidas nos planos de ação formuladas.

- **Revalidação do pacto**: no final, podemos mostrar o pacto da equipe na parede e perguntar a cada um se incluiria ou excluiria algo, para então repactuar "o que queremos" e "o que não queremos", fechando com uma frase de incentivo ao trabalho em equipe, aprendizado contínuo e resultados. Não há e não deve haver um formato único. Devemos nos esforçar para termos sempre um local e uma programação diferente, sempre que possível, de forma a permitir a quebra da rotina e do modelo-mental vigente. Se buscamos inovação e engajamento, é preciso oferecer exatamente isso em nossas retrospectivas. Não queremos apenas mais do mesmo.

Variado & atrativo

- **Local** – Já fizemos em um espaço equivalente a um minimuseu com a história da empresa, em uma sala de reuniões destinado a diretoria, em diversas salas de reuniões do TecnoPUC; sempre que possível faço partes ou o todo ao ar livre, em um terraço ou sob as árvores.

- **Dinâmicas** – Veja em http://www.funretrospectives.com uma infinidade de modelos, técnicas e oportunidades a utilizar, brainstormings ou pequenos treinamentos para introdução de novos conceitos ou reciclagens, o uso de games ou dinâmicas divertidas para fixação de conhecimento ou exemplos para que a equipe reflita e desafie-se a melhorar sempre mais.

- **Games** – Veja em http://tastycupcakes.org/pt uma infinidade de games e dinâmicas de fixação de conceitos ágeis, com duração desde alguns minutos até para mais de uma hora; é só uma questão de usar com sabedoria. Há os de fixação de conceitos de Kanban, de Daily, de planning, de iterações, métricas, User stories; tem de tudo, para todos gostos.

- **Café coletivo** – Uma opção que surtiu um efeito muito legal de integração é realizar um café coletivo cedinho pela manhã, em que cada um

traga alguma coisa e coloque na mesa. É uma oportunidade para os que gostam de fazer em casa um bolo ou biscoitos. Durante o café, aproveitem e coloquem um assunto divertido, para que cada um se posicione. O próprio café é um quebra-gelo.

Quadros com diagnóstico, desejos e planos de ação

Prepare um quadro em que as sugestões, diagnósticos e planos que emergirem durante a retrospectiva fiquem registrados e que depois possam ser mantidos à vista, podendo ser em diferentes formatos. Podemos usar um Scrum Board (*to do-doing-done*), ou um grande quadrado riscado na horizontal e vertical, com quadrantes para visualizar o índice de crença e execução dos itens analisados, com cinco linhas – pessoas, ambiente, ferramentas, método e produto.

A recomendação é que o quadro de diagnóstico, desejos e planos de ação, da forma como foi montado durante a retrospectiva, seja afixado junto às mesas de trabalho do time, para que não seja esquecido. Deve ser usado a favor de um bom ambiente e da melhoria contínua, pessoal, profissional e metodológica em prol de melhores resultados para o ecossistema e negócio.

Na prática, tudo é para trazer todos de corpo e alma para esta discussão, é debater e criar microplanos de ação para todas as oportunidades de melhorias que venham a ser levantadas pela equipe, sem preconceitos ou filtros.

7.4 Melhoria contínua em TI é com Dojos

"O homem que comete um erro e não o corrige está cometendo outro erro."
– Confúcio

Em todo evento ou empresa a que vou, há sempre um falso-dilema: reivindica-se maior qualidade, alinhamento tecnológico, boas práticas, Clean Code, TDD, mas as empresas insistem em queimar dinheiro com cursos e consultorias que não resolvem, pois eles são efêmeros. A solução é muito "treino", a solução é Dojo.

Trata-se de uma analogia ao DOJO KATA, local de treino das artes marciais japonesas, com movimentos de ataque e defesa, no desenvolvimento das

aptidões físicas e psicológicas para o verdadeiro combate, incorporando-os ao seu modelo mental de forma a torná-los parte de seus reflexos naturais, quase instintivos.

No nosso mundo de métodos ágeis, os mais conhecidos são os Coding Dojos usando conceitos de TDD (com testes unitários) e StartUp Dojos para inovação e empreendedorismo, mas há outros, como UX Dojo (*User Experience*), Couch Dojo, Management Dojo e Agile Games direcionados à gestão e princípios ágeis sob diferentes abordagens.

O principal objetivo da realização de Dojos é qualificação, o tipo de Dojo deve ser escolhido de acordo com os princípios e habilidades que queremos fixar ou evoluir. Não deve ser um evento para especialistas; o interessante é termos diferentes níveis de conhecedores e conhecimento, pois é na união de diferentes bagagens e vivências sobre o tema e tecnologia escolhida que obteremos o "mix" que resultará em geração e fixação de conhecimento, inovação e sustentabilidade.

Tenha atenção a alguns pontos chaves, como:

- Ambiente descontraído, divertido e sem pressão por resultados;
- Clima de inovação e empreendedorismo, tentar fazer diferente;
- Relaxem, erros são bem-vindos, fazem parte de processos ágeis;
- A discussão é no campo das ideias, não pessoalize;
- Treinar o ouvir e falar, cada um será aluno e professor;
- Total sentimento mútuo de colaboração e cooperação.

Coding Dojo

Pode ser em BackEnd ou FrontEnd, mantendo foco em Pair Programming e TDD. É uma técnica para a introdução e fixação de conceitos de programação ágil, que visa à produtividade, qualidade, colaboração, primando sempre pelos três pilares do Scrum: transparência, inspeção e adaptação. Veja mais dicas em http://dojopuzzles.com.

1) Até 15 pessoas em volta da mesa de reuniões, um só notebook, conectado ao projetor; inicia-se com uma dupla, sendo um o piloto e outro copiloto;

2) Os dois falam em voz alta o que pretendem, discutem, mas só o piloto pode digitar, sempre deixando claro os seus propósitos, meios e objetivos;

3) Após 5 minutos, o piloto volta a ser plateia, o copiloto assume como piloto e o próximo da mesa assume como copiloto, para que todos experimentem;

4) A plateia não pode sugerir ou questionar as decisões da dupla, mas pode perguntar se não estiver claro o que estão fazendo, não pode ser no escuro;

5) Segue o ciclo de desenvolvimento TDD (*red-green-refactor*), iniciando pela classe de testes, desenvolvendo a classe da solução, validando e refatorando.

Os principais aprendizados são: ouvir sem falar, entender o outro, continuar sem recomeçar etc.

Management Dojo

Técnica criada pelo grande Manoel Pimentel, matadora para a realização de um brainstorming dirigido para a solução de problemas. Usa-se um quadro tamanho A3 com 5 áreas e a dinâmica é de Dojo:

[/list] * Problema: qual é o desafio a resolver; * História: quais são os fatos conhecidos para solução; * Ideias que podem ser úteis, solução; * Hipóteses: que é na verdade o plano de ação; * Métricas para monitorar a execução. [/list]

A dinâmica é semelhante ao Coding Dojo:

1) Iniciamos com 5 minutos de debate em que todos falam e colocam post-its;

2) Seguidos de 3 minutos para só o piloto e copiloto falarem e alterarem;

3) Fazemos vários ciclos de 3 minutos, girando o piloto e copiloto;

4) Podemos permitir alguns tempos de 5 minutos de brainstorming adicionais.

Se esta técnica for usada em um formato maior, semelhante a um Open Space, divide-se em equipes, com mesmo tema ou diferentes, realiza-se o debate e é possível fazer um final muito interessante. Pedimos que fique apenas um de cada grupo e os demais vão cada um para um dos outros grupos, de modo que nos últimos 5 minutos um dos integrantes originais de cada explica para os outros.

UX Dojo

Semelhante ao Coding Dojo. Podemos reunir um mix de profissionais de UX, mas desafiando-os a fazerem uma análise e diagnóstico de uma página, sobre conceito e melhores práticas, construindo *Mock-ups* para entendimento, que são peças em tamanho real ou exagerado do produto, podendo ser desenhos, algo tipo maquetes, simulações ou abstrações:

1) 10' – Apresentação da página ou desafio e características pela PO, SEO e UX;

2) 10' – Debate aberto a todos para dúvidas e melhor entendimento;

3) 05' – Formação das equipes, cada uma com um UX e um convidado;

4) 20' – Cada equipe discute e desenha as suas proposições de melhorias;

5) Ciclos de 05' – Cada equipe tem 5 minutos para apresentar suas ideias;

6) 25' – Debate aberto para montar uma proposta com o melhor de todas;

7) 15' – Avaliação do evento.

StartUp Dojo

A cada ano no Agile Brasil um assunto chama especial atenção: é a trilha SartUp. Sempre há uma profusão de palestras, lightning talks, mãos na massa e muitos debates sobre formatos, modelos, Business Model Canvas, Lean Canvas, em Brasília, Floripa, SP, RJ, ... o de Porto Alegre é organizado pelo amigo Daniel Wildt, um dos grandes nomes da agilidade gaúcha.

1) Lightning Talk sobre o Business Model Canvas;

2) Divisão da galera em equipes entre 4 e 8;

3) Debate para escolha de um negócio a ser detalhado;

4) Decidido o negócio, montagem do Canvas;

5) Pitch do Canvas de cada equipe e perguntas;

6) Lições aprendidas e encerramento.

Agile Games

Tenho convicção de que uma das melhores maneiras de fixar conteúdo sobre métodos e técnicas ágeis é através de Agile Games, quando os participantes são desafiados a simular e exercitar certos conceitos de forma controlada, de modo que seja possível gerar gargalos, riscos, situações em que decisões serão tomadas e logo depois analisadas.

Hackatonas

As hackatonas, "hack-a-ton" ou maratonas de hackers são eventos que reúnem uma galera decidida a construir pequenas soluções ou serviços no intervalo de 1 ou 2 dias. Usualmente são concursos em que os produtos finalizados são apresentados e os melhores dão certo destaque ou prêmios as suas equipes.

Empresas como o Facebook se utilizam deste expediente para mobilizar centenas de seus desenvolvedores, gerando o evento e contando com dezenas de suas equipes em diferentes locais, envolvendo-os em uma atmosfera saudável, divertida, criativa e também competitiva. Ouvi dizer e há registros na web de que o botão "curtir" e outras funcionalidades foram criadas nestes momentos.

Para uma incubadora, aceleradora, grandes empresas ou condomínios, o objetivo é instigar o empreendedorismo e inovação. Para os profissionais, trata-se de oportunidades de experimentar, de ousar e ganhar destaque para si, sua equipe ou para sua ideia, que poderá evoluir e tornar-se um negócio.

Coach Dojo

É uma dinâmica provocativa em que desafiamos cada integrante a assumir o seu e outros papéis durante um Dojo, que se torna quase uma terapia em grupo. É importante construir um pack de situações detalhadas por papel e a cada rodada pedir que cada integrante assuma um perfil.

Na primeira rodada, distribuímos um cartão para cada integrante, contendo cada um a forma como deve se comportar em certo evento ou timebox, para que tenhamos artificialmente situações que acontecem na prática e que eles terão que mostrar como gostariam de endereçar. Por exemplo, uma Daily em que as tarefas estão atrasadas e estamos no início do Sprint:

- Um assume como Scrum Master e deve deixar rolar sem tentar resolver;
- Outro assume como Product Owner e deve pressionar por horas extras;
- Um integrante deve querer fazer as horas extras imediatamente, hoje;
- Os outros integrantes devem atuar por seu entendimento e instinto;
- Outra rodada poderia ser uma retrospectiva em que um integrante está no celular o tempo todo e outro está puxando assunto com mais alguém sobre algo que nada tem a ver com o trabalho.

Pode haver uma Review em que um usuário reclama muito, cada rodada deve ter um tempo hábil de 5 minutos, seguidos de 3 minutos de debate sobre o que aconteceu e qual seria a melhor solução.

7.5 Resumo de 4 dicas em cada momento

"O conhecimento é uma crença verdadeira justificada."
– Platão

A seguir, um resumo com quatro importantes dicas durante o fluxo que perpassa todas as timeboxes e importantes momentos do método Scrum. É uma revisão dos principais pontos de cada um:

Sprint Zero

- Bom Discovery e Sprint Zero são premissas para o sucesso dos Sprints;
- Envolver o usuário, brainstorming, consultas, participação, envolvimento;
- O PO conta com UX, SEO, SQA e deve usar referências técnicas no time;
- Iniciar um Sprint cheio de dúvidas e "a definir" gera frustrações e riscos.

Planning

- Iniciar com visão do RoadMap, objetivo da Release e Sprint, velocidade;
- Relembre as melhorias e decisões trabalhadas na última retrospectiva;
- Use as paredes, mantenha DoD e tudo que foi dito e decidido bem visível;
- Após estimar as US, alinhe Sprint Zero, Pair, P&D, eventos e pretensões.

Daily

- Foco total no atingimento do objetivo do Sprint (riscos e oportunidades);
- Assertividade, cada um deve preparar-se: o que foi, será, rolos e boas;
- O que falar ou não? Um resumo daquilo que é relevante para o Sprint;
- Após 3 Dailys sem decisões, risco, oportunidade, olhe embaixo do tapete.

Quadro (não é timeboxe, mas merece estar aqui)

- Deve claramente refletir o dia a dia do Sprint e apontar suas tendências;
- É um reflexo da organização, comprometimento e engajamento do time;
- Se está defasado, atenção, não entenderam o valor do Kanban nem Daily;
- Quadros, linhas, colunas, post-its, a cada Sprint, tentem melhorar.

Review

- Time Scrum (e Review) não tem dono, evite "ser a estrela da festa";
- Se principais interessados não estão presentes, é só "mais uma reunião";
- Devem estar claros e acordados o roteiro e responsabilidade de cada um;
- Cada integrante deve preparar-se, já que Review gera impressões e vínculos.

Retrospectiva

- O valor é o de melhoria contínua – processo, profissional e pessoal;
- A programação deve estar sempre alinhada com o momento do time;
- Evitar choradeira e resmungação, o prisma é na solução ou melhoria;
- A principal motivação é ter orgulho do que e com quem se faz.

Capítulo 8

Gestão e liderança

Assim como os conceitos anárquicos, a auto-organização gera mal entendidos aos mais afoitos. O anarquismo (do grego `anarkhos`*, "sem governantes") é uma filosofia política que engloba teorias, métodos e ações que objetivam a substituição das formas de governo compulsório por aquelas livremente aceitas. Anarquia significa ausência de coerção e não a ausência de ordem.*

8.1 Esferas de atuação

"Você pode encarar um erro como uma besteira a ser esquecida, ou como um resultado que aponta uma nova direção".
– Steve Jobs

Genba é um termo da cultura japonesa apropriado pela cultura Lean para identificar a importância de entender qual é o lugar e momento em que o

"valor" é de fato criado. Entendamos *valor* como atingimento dos melhores resultados, quer para um debate de ideias, solução de problemas, resolução de conflitos, aproveitamento de oportunidades, no pleno potencial do objetivo.

É um conceito que nos leva a tentar entender algo através da participação ativa, presenciando fatos ao invés de apenas recebendo relatos. É criar ou aproveitar o momento certo, no lugar apropriado e com as pessoas certas, para então tratar de assuntos que necessitam de domínio para serem endereçados.

De nada adianta chegar a conclusões e tomar decisões conversando com as pessoas erradas ou tentar solucionar algo baseado em relatos indiretos ou passando recados, meios imprecisos e passionais. Fazer isso, tendo a oportunidade e acesso aos locais e pessoas corretas é mais que desperdício. É um grande desafio para aqueles mais imediatistas, passionais ou geniosos.

Se somos contra algo, é no fórum adequado que devemos usar de nosso poder de negociação e argumentação para tratar deste assunto. De nada adianta não ter argumentos e apelar para frases de efeito e retórica nos corredores e reuniões. Provavelmente, esta atitude não só não resolverá nada como irá gerar novos e maiores empecilhos para todos, prejudicando o ambiente geral e a execução.

Contrato e execução ágil

Quando um cliente, externo ou interno, contrata um serviço de desenvolvimento de software, estabelecem-se duas esferas claras de atuação: uma de contrato e outra de execução. Não entender o que é uma ou outra leva pessoas a "desperdiçar" seu tempo e dos outros debatendo coisas fora de suas alçadas; pior que isso, gerará tensões e reações que prejudicarão ações futuras, sinergia e empatia.

Fig. 8.1: Esferas de atuação

Um cliente que não entenda o conceito de Genba e pressiona questões negociais no dia a dia com a equipe de desenvolvimento gerará e ampliará cada vez mais um sentimento de autodefesa que provocará um efeito diretamente oposto ao desejado. Ingenuamente, exercer uma pressão abstrata, descontente, sobre questões fora da alçada de qualquer dos interlocutores só leva à frustração e angústia.

Um cliente discutir e ameaçar com cláusulas contratuais os integrantes de um time de desenvolvimento é equivalente a estes integrantes discutirem com ele detalhes da configuração do Eclipse ou problemas salariais. O segredo de um bom relacionamento, empatia e sinergia entre fornecedor, cliente e equipes é Genba, esferas de atuação. É cada macaco no seu galho.

- **Cliente x fornecedor** Deve haver uma política de confiança e transparência, ambos cientes do papel de cada um e de que tipo de relacionamento querem construir. Relações baseadas em artes cênicas e teatralidade inconsequentes geram uma relação prejudicial a ambos, de defesas e ataques.

- **Cliente x contrato** Deve trabalhar para deixar o mais claro possível suas expectativas e necessidade, com clareza nos termos e cláusulas que

regem esta relação. Cada qual deve nomear seus representantes executivos e jurídicos, pois contratos devem refletir boa relação entre as partes e sustentação legal.

- **Cliente x time Scrum** O cliente deve empoderar seu representante, o Product Owner (PO), identificando também seus stakeholders em cada nível e área, de forma que o time junto ao PO tenham máximo entendimento das features e protagonistas, necessidades e oportunidades, para geração de valor.

- **Fornecedor x contrato** Os termos relacionados à metodologia a ser praticada devem ser ao máximo esclarecidos. O cliente deve ser apresentado plenamente aos preceitos e condicionais relacionados aos princípios ágeis e do método que será aplicado, alinhando as condições e expectativas.

- **Fornecedor x time Scrum** É preciso dar as condições e empoderamento ao time para que ele possa realizar seu melhor trabalho, consciente da grade de conhecimentos, capacidades, necessidades e restrições. Cabe ao fornecedor proporcionar ao time condições compatíveis ao contrato e expectativas.

- **Contrato x time Scrum** É importante que o time tenha ciência do que é esperado dele, de como o contrato exprime sua atuação, qual a responsabilidade de parte a parte; mas NÃO é alçada do time estabelecer discussões com o cliente sobre estes temas, o que deve ser redirecionado ao gerente ou área apropriada. A incapacidade de percebermos e termos uma atuação condizente em diferentes esferas gera um custo alto em desperdício, tensão e dispersão. Quanto mais perdemos o foco de nossos interlocutores, mais energia se esvai discutindo questões que não estão maduras ou necessitam outro fórum. O estabelecimento de discussões e pressão na esfera de atuação incorreta gera um custo em confiança e quebra de empatia que, mesmo redirecionando e solucionado na sequência, exigirá ainda mais energia e tempo para resgatar.

8.2 Gestão e liderança ágil

"Se você faz algo de bom e tudo dá certo, acho que é hora de pensar em outra coisa e tentar adivinhar o que vem pela frente."
– Steve Jobs

Equipes auto-organizadas somente conseguem ser auto-organizadas se houver um nível externo que os legitime e apoie, empoderando e gerando o substrato onde poderão fazer realmente um bom trabalho intra e intertimes, negociando os recursos necessários e disponíveis.

Enquanto equipes auto-organizadas têm foco em projetos e resultados, os gestores e a estrutura de apoio da empresa geram as condições para que o esforço de cada uma seja potencializado. Equipes ágeis precisam de suporte organizacional para manter um ritmo construtivo adequado e sustentável a todos os envolvidos, gerando valor à empresa, clientes e a seus integrantes.

Métodos ágeis preconizam gerenciamento de projetos baseados em times pequenos e auto-organizados, com forte visibilidade, adaptação e um processo de desenvolvimento interativo-incremental com foco na entrega do melhor valor ao negócio, no menor tempo possível. A transferência de alçada aos membros de cada equipe os autoriza a tomar decisões internas necessárias que viabilizem seu melhor trabalho/valor.

Auto-organização não pressupõe a eliminação de gerentes; é um movimento que retira as equipes do estado de equilíbrio passivo, aumentando a entropia geral do sistema através de times autodeterminados, com a redução da hierarquia e comando-controle. A auto-organização acontece com gestores proporcionando e apoiando equipes autônomas em uma organização de aprendizado construtivo e coletivo.

Tão importante quanto as equipes em uma cultura ágil baseada em times auto-organizados é o papel do gestor, responsável por criar o substrato que viabilize e proporcione as condições que cada time necessita para fazer um bom trabalho, alinhado ao que se espera dele.

O Management 3.0 de Jurgen Apelo se propõe a oferecer uma visão do papel do gestor e líder na cultura ágil, que vem a cada ano sendo adotada por um número maior de empresas.

- **Energizar as pessoas** Gestores devem incentivar as pessoas a se sentirem motivadas a fazer um bom trabalho, sendo confiantes, proativos à inovação e superação. As pessoas têm que querer e poder construir um ambiente inspirador;
- **Empoderar os times** – Capacitar pessoas e times para que se concentrem na busca pela autonomia em tomar as decisões necessárias para seu trabalho, com conhecimento hábil, habilidade e alçada para defender seu ponto de vista;
- **Alinhar restrições** Uma cultura organizacional aderente à auto-organização exige que o sistema seja claro, transparente quanto ao papel e alçada de cada um. As restrições dirigem a auto-organização na direção certa para geração de valor;
- **Gestão por competências** Disciplina imprescindível nas organizações: monitorar, perceber e promover o desenvolvimento de competências individuais e coletivas para que seja possível atingir objetivos e gerar valor a todo o sistema;
- **Crescimento da estrutura** Cabe ao gestor trabalhar para constituir e fazer crescer uma estrutura que sustente uma cultura ágil, potencializando a comunicação e negociação iterativo-incremental de valor para otimizar os resultados;
- **Melhorar tudo** É importante que o gestor tenha uma visão estratégica de sua área e organização, buscando aprimorar pessoas, equipes, ambiente, recursos disponíveis, apoiando para que a auto-organização gere o máximo de resultados.

É preciso blindar a equipe de discussões contratuais e apenas envolvê-la quando estas questões estiverem esclarecidas, pois um time pressionado irá perturbar sua cadência e foco em valor para questionamentos sobre os quais não tem alçada, mas podem influenciar negativamente suas decisões futuras. Esta situação bem trabalhada chegará à equipe arredondada.

Gestores ágeis são necessários na adoção Scrum e no estabelecimento de uma cultura de melhoria contínua e sustentável. As equipes precisam de apoio

da gestão ao receberem a estrutura e aportes de recursos, treinamento e ambiente para se auto-organizarem de forma efetiva e consistente.

Uma empresa com várias equipes, cada qual com um projeto ou mesmo escalando o Scrum se beneficia exponencialmente com a atuação de um líder inspirador, disposto e disponível, que participe de momentos relevantes e significativos de modo a compreender necessidades e oportunidades em suas equipes e integrantes.

Não é incomum a necessidade de um gestor precisar blindar uma ou mais de suas equipes de forma que pressões e discussões inadequadas se dirijam ao fórum apropriado, não afetando desnecessária e inutilmente, e algumas vezes minando o ambiente saudável e performance de uma equipe ágil.

Demora-se para construir um time ágil. Passamos semanas ou meses desde a etapa de *Forming*, *Storming* e *Norming*, para então chegar ao *Performing* (Curva de Tuckman). É um grande desperdício abalar este crescimento e voltar à etapa de *Storming* por motivos que muitas vezes se esvaziam ou porque não cabia à equipe participar, apenas conhecer o resultado, que normalmente pouco lhe afeta. São meses jogados fora e perder confiança é muito fácil frente a ganhá-la novamente.

8.3 Agile é uma revolução permanente

"Temos o destino que merecemos. O nosso destino está de acordo com os nossos méritos."
– Albert Einstein

Fico inquieto com a estratégia de algumas empresas, pois implantar Scrum é só o primeiro passo, talvez o mais fácil deles. Não podemos ceder ao reducionismo de papéis, timeboxes, Kanban e postits. A revolução se inicia com a metodologia, mas só será sustentável se houver uma mudança cultural sustentável de fato.

A solução não é o método, é o modelo mental. Algo como o que Trotski chamava de Revolução Permanente: de nada adianta trocar a burocracia e obediência aos métodos tradicionais pela burocracia e obediência aos métodos Scrum e Kanban, trocando de protagonista em seus medos e acomodações.

A verdadeira revolução inicia na equipe, é o primeiro passo. A mudança dependerá da ampliação e interação com outras equipes, com a organização como um todo, trabalhando pela constituição de um ecossistema sustentável, em sua relação com inter e intraorganizacional, clientes, parceiros, fornecedores e mercado.

Assim como a visão de Revolução Permanente, vivemos dicotomias temporais em nossas empresas. Ao invés de mudanças intrínsecas como Japão, EUA e Europa, temos mudanças extrínsecas em que adotamos Agile por motivos e condições que o tornam distorcido, oportunista e processual, não cultural.

Transparência, inspeção e adaptação são os pilares do Scrum, mesmo assim, em função da dicotomia existente no entendimento do que é agilidade e o que é a auto-organização, entre teoria e prática, muitos optam por distorcer seu dia a dia com muros e trincheiras, justificando que se fizerem diferente poderão se expor e, se algo der errado, serão cobrados por isso.

Espiral de aprendizados

O primeiro passo é um piloto; é gerar resultados, provocar comparativos entre o modelo tradicional e o ágil, com acertos e erros. Um ou mais pilotos vão exercitar técnicas e percepções, algumas em um mar de novas práticas, mais colaborativas.

Mas o *Go – No Go* entre pilotos e *rollout* não é o fim da história, é apenas o primeiro passo da estrada. Entender que a mudança será uma constante evolutiva a cada fechamento de Sprint, ou seja: uma Revolução Permanente. A zona de conforto sempre será um canto da sereia a nos chamar.

O próximo passo são comunidades de prática intraorganizacionais, escalando-as de forma que grupos de PO, SQA, Dev, SM, UX, SEO etc. compartilhem, interajam com seus aprendizados vicários, disseminem boas práticas, inquietações, simulações, problemas e soluções. Isso mantém todos ligados na Revolução Permanente, relembrando-nos a cada Sprint que sempre estaremos na estrada.

Se estamos trabalhando em uma cultura Kaizen, de melhoria contínua auto-organizada, com timeboxes específicas para exercitar com os stakeholders a cada Sprint Review e internamente ao time a cada Sprint Retrospective.

O restante cabe à aplicação de modelos de gestão do conhecimento, privilegiando maior interação interequipes, compartilhando e consolidando o aprendizado adquirido a partir da experimentação

Conclusão

Já tivemos grandes nomes, pensadores, pesquisadores, intelectuais, todos falando e tentando entender o que é a cultura organizacional, formada de microculturas e inserida em culturas regionais e nacionais. Por mais que leiamos e escrevamos, sempre teremos muito o que ler e aprender sobre pessoas e suas relações.

Não é um desafio trivial, mas tudo se inicia na consciência de que é um desafio cotidiano e que a natureza humana tenta voltar inconscientemente à sua zona de conforto, não por opção ou por ser melhor, mas porque já é conhecida. Uma adoção ágil sem mudança cultural pode trocar o comando-controle explícito por um implícito, velado no sucesso e muitas vezes ruidoso frente a erros.

Agilidade é uma Revolução Permanente, não no sentido literal, mas figurativo, pois sempre teremos o que melhorar, queremos sempre ser mais felizes e também orgulhosos do que fazemos e com quem fazemos. Nada mais apropriado à TI, pois na nossa área não dá para marcar touca, estamos sempre aprendendo, evoluindo, experimentando.

8.4 Evite ser ágil só enquanto tudo dá certo

"Quem escuta colhe, quem colhe semeia."
– Joseph M. Juran

Evite ser uma empresa, gestor ou profissional ágil só até um erro acontecer, pois alguns nesse momento esquecem tudo o que viram, fizeram e aprenderam nos últimos meses e começam a buscar explicações não baseadas nos debates colaborativos e decisões colegiadas, mas só o que importa é achar um culpado.

Praticar agilidade é antecipar riscos e oportunidades, é aprendizado diário, é tirar o tapete da sala. Não quer dizer que não haverá mais poeira; apenas

que ela vai ficar visível o quanto antes porque não terá tapete para varrer para baixo dele, mascarando problemas por meses. Em equipes ágeis, a transparência e coletivo são o grande trunfo. É frente a problemas que este *mindset* se põe à prova!

Por exemplo, uma vez feito o planning, debates instigantes, histórias fracionadas em tarefas, estimativas consensuadas, Kanban e Daily, tudo isso não garante que não ocorrerão imprevistos, e é quando eles ocorrem que separamos o ágil do comando-controle. Ao invés de focar na busca de culpados, é preciso analisar passo a passo tudo o que fizemos, ver por que não percebemos antes o risco de aquilo acontecer, onde precisamos melhorar, sacudir a poeira e seguir.

Gestores

Os melhores desenvolvem novas competências, reinventam-se, assumem nova postura, fazem cursos de Management 3.0, energizam e empoderam seus times, compartilham com eles acertos e erros. Partem da própria responsabilidade em selecionar seus liderados, de alocá-los da forma adequadas, apoiá-los, desenvolvê-los, tornam-se líderes ágeis e inspiradores, delegam de fato e dão suporte.

Contudo, há gestores que curtem os métodos ágeis, mas fazem questão de prestar atenção somente às frases que dizem que haverá entregas frequentes, que a meta é o máximo de valor, eliminação de desperdícios, que o cliente participará mais e compartilhará com o time cada decisão. Fazem que se esquecem de que não é bala de prata, que não é mágico, que é uma estrada e não um passo.

Os resultados são tão eficientes que esquecem como eram seus projetos antes; após alguns Sprints de sucesso já assumem que todos têm a obrigação de dar certo. Erros que antes geravam grandes prejuízos agora acontecem e são mitigados o quanto antes, mas mesmo assim volta a ser importante ter um culpado.

Clientes

Há clientes que reinventam seus contratos, estabelecem relações de confiança e, participando mais intensamente dos projetos, acabam por deter maior

controle sobre a otimização de recursos para entregar o máximo de valor a cada iteração.

Alguns clientes entram nesta vibe Ágil resistentes, mas rapidamente percebem que o maior beneficiado são eles. Então começam a apoiar, participar mais, mas alguns veem riscos em ter uma ampla colaboração e ter que assumir riscos junto; preferem a Lei Ricupero: quando é bom, é nosso, quando é ruim, é de vocês!

Equipes

Por incrível que pareça, muitas vezes o gestor e o próprio cliente entendem que estarão experimentando e aprendendo, que erros são um incômodo, mas uma vez acontecendo, o nosso objetivo deve ser convertê-los ao máximo aprendizado para que não voltem a acontecer.

Mas frente ao erro temos integrantes que inconscientemente buscam achar um culpado que livre todos da sensação de algo ter dado errado e que todos estão juntos nessa viagem. Às vezes, o gestor está ciente e o cliente também de que todos tentamos acertar e que algo passou e nos ensinará; mas tem quem surte e queira a cabeça de alguém, desde que não seja a sua.

A mudança para uma cultura Ágil já é um grande desafio com todos puxando para o mesmo lado. Alguém voltar para sua zona de conforto é pior, mas basta termos consciência de que isso também faz parte e de que precisamos todos manter a coerência e relembrar os quês e porquês. O que não pode é cada vez que um surtar os outros irem na onda.

Tenhamos sempre consciência dos passos, de cada momento, debates e decisões, cada ação na tentativa de acertar, cada contingência e tratamento de riscos. Se algo deu errado é preciso focar no aprendizado antes mesmo de ações e replanejamentos.

Afinal, curso de Agile não é um curso de magia com o Prof. Dumbledore. A proposta é trabalhar projetos complexos de forma a antecipar riscos e oportunidades, o resto é aprendizado.

8.5 Ensaio sobre estimativas

"A qualidade não acontece por casualidade, ela deve ser planejada e executada."
– Joseph M. Juran

Vamos supor que você foi a uma oficina mecânica e pediu um orçamento para arrumar uma batida lateral no seu carro. O dono da oficina diz que avaliou os danos e que "estima" que seria preciso trocar as peças X e Y, chapear a lateral e pintar; mais mão de obra e o custo fica em 10 mil. Ele adverte que este orçamento é fruto de observação e experiência, mas ao desmontar o veículo poderia surgir algo mais, desapercebido.

Ao desmontar a lateral, ele percebe que houve uma torção na longarina e que precisará tracionar de forma a desentortá-la, mas esse é um trabalho oneroso, que envolve usar dois de seus profissionais por uma tarde e isso aumentaria o custo do reparo para 15 mil.

Neste cenário, temos diferentes tipos de oficinas e de clientes, esperamos de ambos uma atitude transparente, ética e consciente. O correto seria a oficina e o cliente terem consciência do que é a responsabilidade de cada um, direitos e deveres, em uma relação de confiança – transparência, inspeção e adaptação.

#1. A oficina lhe comunica antecipadamente

Assim você pode verificar se pode pagar neste momento, pode negociar parcelamento, pode ir lá e pedir para ver o dano na longarina, pode optar por levar em uma oficina especializada, pois uma longarina torta compromete o desempenho do carro e gera riscos de acidentes, há oficinas especializadas.

O dono da oficina não bateu o seu carro, você bateu, ele advertiu que somente teria confirmação quando desmontasse o veículo para ver melhor a extensão do dano. Não adianta descontar nele, afinal, o carro é seu, ele está lhe prestando um serviço e orçar um concerto é sempre passível de ver mais coisas ou não ao abrir.

Por outro lado, se você acha que ele o está enganando, se não tem confiança nele, então não deveria ter levado o carro lá, oficina mecânica deve

ser uma relação de confiança. Deve ter boas referências e deve aparentar e demonstrar ser capaz, confiável e ética.

#2. A oficina faz tudo sem lhe comunicar

Essa atitude provavelmente vai gerar muita discussão, podendo até mesmo terminar em litígio entre as partes. O cliente negociou algo e a oficina não poderia fazer diferente sem lhe comunicar antecipadamente. Isso só pode acontecer se houver um acordo prévio entre o cliente e a oficina, que pode ser baseado em um percentual de variação, intervalo de valor ou quando os mecânicos se antecipam e geram cenários possíveis, mais e menos otimistas quanto aos danos. Se por um lado é importante haver confiança, por outro é preciso manter-se uma boa relação comercial. É papel da oficina utilizar sua experiência no assunto para antecipar as possibilidades ao cliente, por outro lado, é papel do cliente manter-se informado a cada passo para não haver surpresas e ter expectativas realistas.

#3. A oficina realiza só o concerto orçado

Esta atitude é desonesta, pode gerar um acidente e um processo na justiça, pois ficará a dúvida (e terá que ser provado) de que o problema não foi gerado pela oficina, mas pelo acidente anterior, o que pode colocar a responsabilidade sobre a oficina por ter feito o concerto e liberado um carro torto.

O mínimo que se espera de uma oficina é que, ao perceber que o carro está com a longarina comprometida, comunique o dono da real situação do veículo. Daí em diante, a decisão é sua. Não é uma atenuante para a desonestidade caso ele o conheça e saiba que é truculento e errar o orçamento lhe trará dor de cabeça.

Rolar com a barriga e não ser transparente é a pior solução, pois tem perna curta e depois o problema terá se potencializado e virado um problemão. O melhor é enfrentar a situação, repassar a alçada a quem for de direito (dono do carro), que tomará a decisão e assumirá as consequências.

A oficina e o software

Estimativa de software é mais ou menos a mesma vibe. Somente quando a gente começa o serviço é que podemos perceber detalhes técnicos que podem exigir mais tempo e dinheiro. Quanto mais experiente e confiável a "oficina", melhor, mas se a estimativa inicial for um contrato imutável, então a solução é multiplicá-la por dois para garantir que não se vai ficar no prejuízo e pagar para trabalhar. Estimativa identifica um cálculo aproximado, avaliação ou conjectura, é uma aproximação, pressuposto, hipótese ou suposição. Só há uma forma de acertar sempre as estimativas, bastaria calcular um número e colocar uma margem generosa. Quando pressionados para acertar sem errar nunca, é o que fazemos.

8.6 Sem confiança não existe agilidade

"Quer você acredite que consiga fazer uma coisa ou não, você está certo."
– Henry Ford

É preciso que cada stakeholder e cada integrante do time estejam realmente comprometidos com suas decisões e com os princípios que sustentam o método. Tudo parte de estabelecer um ambiente de confiança, com realismo e transparência.

Desenvolvimento de software envolve não só uma linguagem de programação, mas intensas relações humanas, arquitetura de dados, várias frameworks e camadas, consumindo serviços, envolvendo questões de segurança, de persistência, negócio, patterns, apresentação; nada é trivial.

Antes de acusar, voltando à zona de conforto na busca por um culpado, lembre-se que você é parte do projeto e que há responsabilidades e alçadas. Mais importante que não errar é aproveitar os ciclos curtos das iterações para aprender o mais rápido possível. Errar não é de todo ruim, aprenderemos com cada um deles, não podemos é cometer os mesmos erros.

Isso não quer dizer que pessoas não possam ser trocadas, substituídas, mas é preciso que se tenha conhecimento do perfil de cada um, conhecimento, experiência, cognitivo, habilidades, atitude e crenças em um trabalho coletivo. Acima de tudo, agilidade é suporte, aprendizado, é gestão do conhecimento e gestão por competências. Se não for assim, não é ágil, é engodo!

Gestores e equipes

O principal responsável pelo perfil adequado ou não da equipe são seus gestores. É inconcebível que frente ao primeiro erro o gestor se isente de sua responsabilidade na composição e condições do time de fazer bem o seu trabalho. O bom gestor conhece suas equipes, monitora e dá apoio e suporte.

Gestores e Scrum Master

Escolher alguém para SM contra a sua vontade ou sem que ele tenha tempo para este papel é o mesmo que não ter ninguém, é um papel de bastidores. No início, ser facilitador e organizar as timeboxes exige dedicação e planejamento, até virar hábito.

Cliente e PO

O cliente deve escolher seu Product Owner tendo a confiança que ele terá apoio e espaço para fazer o seu melhor. Este é o papel mais importante no método, é quem toma as decisões pertinentes ao que é valor, ROI e aceitação.

Time = PO, SM e equipe

Todos os integrantes de um time Scrum são colegas de jornada, o que quer dizer que o suporte horizontal deveria ser igual para o aprendizado e crescimento de todos, entre desenvolvedores, testador e também com o PO.

Cliente e terceiros

Se funcionários e terceiros alimentarem desconfianças e desacordos essenciais e diferenciados, é preciso que a gestão interfira e resolva, se necessário pela troca do profissional que precise ser trocado, pois enquanto equipe é preciso que todos tenham a mesma alçada e protagonismo.

Conclusão

Inexiste agilidade sem um clima de confiança entre as partes. Se uma parte não retribui ou não possui os atributos desejados, nada pode ser uma surpresa para ninguém; é preciso que haja continuo feedback, transparência.

Somente assim e com o suporte preciso é que surge uma cultura de aprendizado.

Um curso Scrum não muda pessoas, apenas as instrumentaliza para que empreendam a mudança um passo de cada vez. Agilidade é um processo de mudança intrínseca que com boa vontade demandará semanas ou meses.

8.7 Contratos ágeis

"Se quiser ter uma boa ideia, tenha uma porção de ideias."
– Thomas Edison

A adoção de métodos ágeis não deve se restringir a equipes de tecnologia, pois para atingirmos a plenitude do seu potencial, na valorização das pessoas, da interação entre elas, na entrega anteci¬pada e diária de valor ao negócio, é necessário buscar o envolvi¬mento de todo o ecossistema, com os players internos e externos.

Contratos waterfall e método tradicional

No gerenciamento tradicional, tínhamos custos, cronograma e escopo, que deveriam ser minuciosamente conhecidos e rigidamente gerenciados, lançando mão do famoso processo de *change management*, para que a cada pequena alteração iniciasse um ciclo de gestão de impacto e custos.

Era um jogo sutil de opacidade, com falta de transparência de parte a parte, posto que cada lado exercitava sua vontade de ganhar o máximo e perder o mínimo. Isso acabava em acusações, desgastes e desperdício, salientando que algo sempre era entregue e o usuário pagava a conta.

O insucesso era decorrente do descontrole de interesses, falta de transparência (de verdade) e valorização excessiva do contrato com dispositivos legais para vantagens e penaliza¬ções ($), tanto cliente quanto fornecedor. Após um tanto de desgaste, o produto acabava em segundo plano.

Contrato ágil de desenvolvimento

Método ágil **não** resolve os problemas, ele apenas garante que os problemas e soluções serão compartilhados. Se não houver convergência e tanto

cliente quanto fornecedor quiser ganhar do outro, **não** tem maneira que faça dar certo!

Um contrato ágil não é anárquico, tem um escopo desejado, valores a serem cobrados e tempo combinado. A diferença é que sua execução resguardará o método, timeboxes, papéis e artefatos, de forma que um processo ágil de sucesso possa ser executado e, ao final, o melhor produto seja entregue em comum acordo.

A partir do primeiro dia, temos duas claras esferas de atuação: uma **com** crachás, exercido pelo gerente de conta e aquele designado pelo cliente para questões financeiras; a outra é **sem** crachá, em que **todos** serão um só time, com transpa¬rência e atitude, sentimento de pertencimento e foco em agre¬gar valor.

Case hipotético

O site de uma revista é composto por edições, capa, maté¬rias, vídeos, mural, enquetes, galeria de fotos, blog, votações e agenda. Todas essas funcionalidades estarão explicitadas superficialmente no contrato, detalhadas o suficiente para entendimento do que são e características úteis às mesmas. A partir deste entendimento, há um trabalho de estimativa em alto nível, dimensionamento de equipe e tempo.

- Contratação de uma equipe de 10 pessoas por 5 meses
- Custo mensal de 100 mil, total de 500 mil
- Product backlog explicitado
- Discovery e Delivery com Sprints quinzenais

Se há confiança e colaboração, há a certeza (independen¬te do crachá de cada um do time) de que as decisões serão tomadas de forma colaborativa e responsável, visando transformar o resultado do projeto em um case de sucesso para o negócio e que alicerçará visibilidade e novos contratos.

A premissa usada para equipes internas é a mesma na contratação de fornecedores. Devemos ter orgulho daquilo que fazemos, com quem fazemos e porque fazemos, com valores e culturas ágeis e construtivistas.

Capítulo 9

Outros métodos

O Scrum é um framework dedicado à gestão de projetos ágeis, mas é do Lean que absorvemos princípios e técnicas de estratégia e uma visão mais ampla; vêm do XP técnicas importantes de construção e engenharia ágil; do BDD os conceitos de User stories e linguagem ubíqua, única entre time e todos os demais interessados, baseados em conceitos de negócio e não técnicos.

9.1 eXtreme Programming (XP)

"Na mente dos principiantes existem muitas possibilidades, enquanto na mente de especialistas elas são poucas."
– Shunryu Suzuki

Em meados da década de 90, Kent Beck, Ward Cunningham e Ron Jeffries lançaram http://www.extremeprogramming.org. O Scrum e a XP são

complementares, boas práticas de gerenciamento e de engenharia de SW. O nome vem da alusão ao uso de boas práticas e valores ao extremo, como:

- Se revisar código é bom, revise o tempo inteiro, programe em pares;
- Se testar é bom, todos testam o tempo todo, unidade e funcionais;
- Se o projeto é bom, refatorar será parte das funções diárias de todos;
- Se ter simplicidade é bom, devemos ter o projeto mais simples possível;
- Se arquitetura é importante, todos trabalham para definir e redefini-la;
- Se teste de integração é bom, vamos integrar e testar diariamente;
- Se iterações são boas, serão feitas na escala de minutos e horas;
- Cliente sempre disponível, para dúvidas e repriorizações cotidianas;
- Adotar padrões de codificação em consenso, o mais simples possível;
- 40 horas semanais de trabalho; descansar garante energia e ideias.

Aplicar valores e princípios durante o desenvolvimento:

- Jogo de planejamento (*Planning Game*);
- Pequenas versões (*Small Releases*);
- Metáfora (*Metaphor*);
- Projeto simples (*Simple Design*);
- Time coeso (*Whole Team*);
- Testes de aceitação (*Customer Tests*);
- Ritmo sustentável (*Sustainable Pace*);
- Reuniões em pé (*Stand-up Meeting*);
- Posse coletiva (*Collective Ownership*);

- Programação em pares (*Pair Programming*);
- Padrões de codificação (*Coding Standards*);
- Desenvolvimento orientado a testes (*Test-Driven Development*);
- Refatoração (*Refactoring*);
- Integração contínua (*Continuous Integration*).

Existem seis fundamentos da XP que considerei elucidativas:

- Distinguir as decisões do negócio e a alçada da equipe;
- Escrever testes unitários antes de programar e manter executando sempre;
- Praticar Pair Programming, um pouco a cada dia;
- MVP em produção e seguir na direção que se mostra mais favorável;
- Integrar e testar todo o sistema várias vezes ao dia;
- Começar simples e refatorar, mais flexibilidade e menos complexidade.

9.2 Lean Software Development

"A solução mais simples é geralmente a melhor solução."
– Ken Schwaber e Jeff Shuterland

Em torno de 2003, o casal Mary Poppendieck e Tom Poppendieck promoveu o método Lean para desenvolvimento de software, baseado nos princípios do Lean Toyota Production System, com foco na eliminação de desperdícios, velocidade de processos, qualidade e cadeia de valor.

O termo Lean Software Development teve sua origem em 2003 com a publicação do livro homônimo de Tom e Mary Poppendieck, quando apresentaram como aplicar seus princípios no desenvolvimento de software: *"Ter as coisas certas no lugar e hora certa, desde a primeira vez, enquanto se elimina o desperdício, estando sempre aberto a mudanças."*

Desperdício

É tudo aquilo que não agrega valor ao cliente, como: trabalho incompleto (artefatos inacabados); burocracia ou gerenciamento desnecessário; funcionalidades a mais (complexidade desnecessária); troca de tarefas (código novo e manutenção); redução de *handoffs* (troca de responsabilidade); atrasos (decisões a cada 15 minutos e contornos); defeitos (TDD é investimento).

Inclua a qualidade no processo

Não deixe os testes para o final, evite-os desde antes do início. *"Inspecionar para prevenir defeitos é bom; inspecionar para encontrar defeitos é desperdício."* — Shigeo Shingo.

Crie conhecimento

Incentive a Gestão do conhecimento, tácito e explícito, incentivando através de programas e eventos o seu compartilhamento e geração contínua.

Adie comprometimentos (*last responsible moment*)

Tentar antecipar decisões é um erro, estas devem ser tomadas o mais adiante possível, com informações mais precisas com o andamento do projeto.

Entregue rápido

"A moral da história é que devemos encontrar uma maneira de entregar software de qualidade tão rápido, que nossos clientes não tenham tempo de mudar de ideia." — Mary Poppendieck.

Respeite as pessoas

"A verdadeira inovação da Toyota é sua habilidade em usufruir da inteligência dos trabalhadores comuns." — Gary Hamel

Otimize o todo

É preciso estar atento a todo o processo e as causas que devem ser trabalhadas para a melhoria, com foco em métricas adequadas, conhecimento da

cadeia de valor e da satisfação do cliente.

9.3 Kanban

"Qualidade é responsabilidade de todos."
– W. Edwards Deming

O termo Kanban significa "visual" e "cartão", identificando um sistema de informação que controla a fabricação de produtos conforme a quantidade e tempos necessários em cada uma das etapas de produção, internamente a uma empresa ou mesmo em processos relacionados a fornecedores e clientes.

A origem do termo Kanban se deu no Sistema Toyota de Produção, também conhecido como Lean Manufacturing, artefato e técnica. É fundamental para o conceito de produção puxada, onde há um esforço para a redução máxima de estoques, de matéria prima, inacabados e acabados, eliminando o desperdício através de uma produção contínua e entregas em tempo real.

Para evitarmos os estoques intermediários, cada equipe deve entender sua velocidade e suprir a próxima etapa do processo de produção de acordo com a velocidade puxada por eles. Este princípio é suportado pelo conceito de WIP ou *Work In Progress*, que especifica a velocidade máxima em determinada etapa, para não sobrecarregar a próxima.

Kanban foi idealizado e implementado com sucesso na indústria, mas sua gestão visual é utilizada em desenvolvimento de software, em áreas de negócio, corporativas, serviços e onde mais houver interação entre pessoas que tenham seu trabalho ou ações de alguma forma inter-relacionadas. Podemos listar alguns objetivos e características do Kanban, com os quais é possível entender porque a maioria das áreas, não só a fabril e de software, vem utilizando a gestão visual em seus processos de produção, serviços e backoffice:

- Minimizar inventário e estoques de insumos, inacabados e acabados;
- Minimizar a movimentação dos materiais em processamento;
- Reduzir o tempo total de produção (*lead time*);
- Evitar gargalos ou carências no fluxo entre etapas do processo;

- Descentralização e auto-organização no controle de produção e estoque;
- Oferecer melhor tempo de resposta a mudanças na demanda;
- Execução em pequenos lotes, tanto quanto possível;
- Manter uma efetiva gestão visual em todas as etapas do processo;
- Garantir a cadência e o controle puxado do sistema, sem interrupções.

9.4 BDD, Behavior Driven Development

"A mudança não é o inimigo, antecipe as mudanças. Elas estão lá para fazer tudo mais flexível."
– Mary Poppendieck

Dan North lançou o BDD em torno de 2003, desenvolvimento guiado pelo comportamento. É uma técnica que encoraja a colaboração entre desenvolvedores, setores de qualidade e pessoas não técnicas ou de negócios em um projeto de software.

Em 2010, North e amigos publicaram sua experiência com o framework RSpec no livro *The RSpec Book: Behaviour Driven Development with RSpec, Cucumber, and Friends*. Desenvolvedores que utilizam BDD usam língua nativa em combinação com a linguagem técnica, o que permite que eles foquem em **por que** o código deve ser criado, em vez de **como** será criado. Assim, BDD pode ser considerada uma *ubiquitous language* ("Linguagem Onipresente"):

- Tudo é comportamento, logo, TI e negócio devem entender e expressar-se sobre o sistema da mesma forma, através de User stories.
- Se tudo é comportamento, tudo no sistema deve sempre expressar e confirmar seu valor para o negócio.
- Detalhar e desenvolver de fora para dentro, de forma iterativo-incremental, apenas o suficiente em cada iteração.

O principal ganho no uso de linguagem ubíqua é incentivar que o cliente ou área de negócio interaja normalmente com os analistas de negócios, a equipe de qualidade e de desenvolvimento, mantendo uma linguagem não técnica, que se utiliza do prisma do usuário e valor percebido por ele.

O formato *Dado* | *Quando* | *Então*, proposto pelo BDD para critério de aceitação de User stories, é uma linguagem flexível e poderosa, que permite a definição de cenários de forma fácil e possibilita executá-los automaticamente, agilizando o feedback e análise dos resultados.

BDD trata-se de um processo colaborativo para a construção de cenários principais e alternativos, uma técnica para entrega de requisitos na forma de executáveis. Com BDD, é possível escrever cenários em um formato que permitirá que sejam executados automaticamente para verificar se o software se comporta conforme o desejado.

O vocabulário que pode ser usado em cenários é definido pelo que a técnica chama de DSL ou *Domain Specific Language*, uma exigência para a automação através de ferramentas BDD, como JBehave para linguagem Java. Ele analisa os cenários a partir de arquivos de histórias, executando-os com testes unitários em JUnit e gerando relatórios. JBehave suporta Maven e é possível configurar a pasta de saída para usar Gradle.

A equipe JBehave disponibiliza um plug-in para Jenkins como servidor de integração contínua. Com a instalação do plugin xUnit e JBehave em Jenkins.

9.5 DDD, Domain Driven Design

"Você é a melhor versão de si mesmo quando você consegue se divertir fazendo o seu trabalho."
– Chris Flink (IDEO)

Domain Driven Design ou Projeto Orientado a Domínio origina-se de um livro escrito por Eric Evans, que apresenta um grande catálogo de padrões, regras de três partes que expressam a relação entre determinado contexto, problema e solução.

Ao usar DDD, nós nos focamos em criar modelos de um domínio de problema. A persistência, interfaces de usuário e mensagens podem vir mais

tarde, é o domínio que precisa ser entendido como diferencial competitivo, distinguindo uma empresa de seus concorrentes.

Modelo de um domínio não são apenas diagramas, mas um conjunto de conceitos necessários para serem implementados pelo software. DDD defende que os especialistas do domínio e desenvolvedores se comuniquem usando os conceitos dentro do modelo, formando assim uma linguagem ubíqua.

Outro ponto importante no conceito de DDD é o contexto onde determinado modelo de domínio existe, usualmente inferido a partir do conjunto de usuários finais que utilizam o sistema. Esses usuários se relacionam com os conceitos do modelo de uma forma singular, para quem sua linguagem ubíqua faz sentido, algo identificado como *contexto delimitado* (DC).

O fato de poder haver mais de um contexto delimitado exige que o DDD se preocupe com a relação e convergências entre estes. Para tanto a técnica propõe um espectro de cooperação entre eles através de seis quesitos:

- *Published language* – linguagem comum para interface;
- *Open Host Service* – protocolo para uso de serviços;
- *Shared Kernel* – uso de uma biblioteca comum;
- *Customer/Supplier* – um consome e é parte interessada;
- *Conformist* – consome mas não é parte interessada;
- *Anticorruption Layer* – uso de adaptadores.

Ao considerarmos a arquitetura de determinado sistema, espera-se que com o uso de DDD só estejamos realmente preocupado com a camada de domínio, que provavelmente não tem muito a dizer sobre as outras camadas apresentação, aplicação ou persistência:

- Apresentação – Aplicação – **Domínio** – Persistência

9.6 PMBOK

"Não permita hierarquia e status nas suas equipes e local de trabalho, eles irão destruir a colaboração."
– Perry Kleban (Timbuk2)

A gerência de projetos tradicional é baseada nas dez áreas do PMBOK, que oferece décadas de experiência em um framework que é praticado em larga escala por governos, grandes empresas, em especial por outras áreas de conhecimento onde ele ainda é o único e melhor meio de gerenciar projetos.

O guia PMBOK possui dez áreas de conhecimento e 47 processos. Muitos profissionais que hoje atuam com métodos ágeis negam a possibilidade de haver qualquer ponto de contato entre estes dois modelos, mas conhecer os principais frameworks e abordagens só tem a agregar.

As dez áreas de conhecimento do guia PMBOK sobre gerenciamento de projetos são:

- Gestão de Integração
- Gestão de Escopo
- Gestão de Tempo
- Gestão de Custos
- Gestão de Qualidade
- Gestão de RH
- Gestão de Comunicação
- Gestão de Riscos
- Gestão de Aquisições
- Gestão de Envolvidos

Em 2013 foi lançado pelo Fabio Cruz o livro *SCRUM e PMBOK, unidos no gerenciamento de projetos* (Brasport), que propõe um modelo híbrido, em que

há ao redor de um time ágil Scrum um gerente de projetos, pela manutenção de diversos planos e artefatos exigidos formalmente em contratos baseados no PMBOK.

9.7 Engenharia de software

"Dar o exemplo não é a melhor maneira de influenciar os outros, é a única."
– Albert Schweitzer

A engenharia de software é uma disciplina para estudo e planejamento para desenvolvimento e manutenção de software.

OOP

Programação orientada a objetos não é um método, mas de nada adianta ter um bom método e processo desenhado sem saber programar direito, afinal software se faz com programas que fazem a coisa certa do jeito certo, com qualidade e permitindo fácil manutenção.

OOP é um paradigma de programação baseado em classes de objetos, contendo atributos e métodos, que uma vez instanciados podem ou não ter direitos a acessar e modificar atributos de outros objetos aos quais estão associados.

Da linguagem SmallTalk da década de 70 até as mais utilizadas linguagens orientadas a objetos de hoje, temos nativas ou multiparadigmáticas, suportando seus conceitos em maior ou menor grau junto a características procedurais.

Algo que serve para reflexão aos mais afoitos que acreditam que isso não é problema: pergunte-se o quanto utilizam de forma consciente e organizada bons design patterns, abstração, herança e especialização e encapsulamento ou o quanto é gerado código espagueti não reutilizável.

Componentização

Há uma disciplina dedicada à engenharia de software baseada em componentes, que enfatiza a separação de domínio no que diz respeito a cada funci-

onalidade necessária em um projeto de software. É uma abordagem baseada na reutilização de componentes independentes de baixo acoplamento.

É uma prática intimamente ligada a bom software com boa relação de retorno sobre o investimento, tanto a curto e longo prazo. Podemos ter componentes na forma de Web Services até arquiteturas orientadas a serviços (SOA).

Uma grande fonte de desperdício que temos na maioria das empresas é redundância de código e alto acoplamento, pois, preocupadas com prazos, as empresas aceitam aumentar indefinidamente o débito técnico gerado por código de baixa qualidade.

MVC

O conceito de desenvolvimento em três camadas conhecido por *Model-View-Controller* gera três camadas independentes, respectivamente para a lógica e regras de negócios, para a apresentação e uma de controle responsável pela intermediação entre as duas primeiras.

O objetivo é a reusabilidade de código, separação de conceitos e facilidade para refatoração ou reconstrução de uma das camadas com mínimo impacto nas outras duas.

Um exemplo é uma aplicação web mudar toda a camada de apresentação, oferecendo uma nova experiência do usuário, ainda usando as mesmas camadas de modelo e controle.

Banco de dados

Em métodos ágeis, com entendimento e construção iterativo-incremental de histórias do usuário, um grande desafio é pensar diferente a forma como modelamos nossos bancos.

Assim como a cultura DevOps, modelagem ágil de banco de dados exige uma mudança cultural e quebra de paradigmas, aproximando os DBAs dos desenvolvedores, trabalhando o modelo de dados de forma iterativo-incremental.

É importante conhecer as diferentes técnicas de modelagem e desenvolvimento da camada de persistência e dados, quer com o SQL embutido no código-fonte, no encapsulamento em classes de dados ou no uso de um framework.

Capítulo 10

Ecossistema

Somos muito mais que as partes separadas. É fundamental que haja um esforço de compartilhamento de conhecimento e modelo mental com todos os envolvidos em todo o ecossistema onde estamos inseridos, fortalecendo uma linguagem ubíqua e entendimento em toda a cadeia, inclusive de áreas corporativas menos envolvidas e outros players de interesse no mercado.

10.1 Execução!

"Algo só é impossível até que alguém duvide e acabe provando o contrário."
– Albert Einstein

Se você chegou até este capítulo é porque já entendeu que é trabalho duro: auto-organização, transparência, inspeção e adaptação na escala de horas, eliminar zonas de conforto e esferas de poder. Não é fácil!

O contexto de agilidade não finda em uma equipe ágil, mas pressupõe que toda a cadeia (ou ecossistema) estará comprometida, de forma que todos os interessados diretos e indiretos entendam a ideia de colaboração, entregas iterativo-incrementais, valor, sustentabilidade e tudo o mais.

O excelente livro *Execução* de Ram Charan e Larry Bossidy esclarece o papel das lideranças em meio ao processo de mudanças para a obtenção de resultados maiores e melhores. Eles apresentam no livro 4 disciplinas:

Estratégia, pessoas, operações e execução

O livro pressupõe as mesmas premissas ágeis de transparência, inspeção e adaptação, além de um capítulo dedicado aos sete comportamentos essenciais do líder – conhecer seu ecossistema, realismo, clareza de objetivos, praxis, meritocracia, orientação e autoconhecimento.

Essencialmente, afirma que mesmo tendo a estratégia certa, pessoas capacitadas e método correto, é preciso "executar". As lideranças e liderados devem ter clareza do que acontece, realizando os ajustes necessários ao que deve ser alterado para que os objetivos sejam superados:

- Execução é uma disciplina e parte integrante da estratégia, uma competência a ser adquirida, estudada, revisada e melhorada;
- Execução deve ser um elemento-chave da cultura da organização, não é responsabilidade de uma pessoa ou área em especial;
- Execução não é um processo de auditoria, é um processo iterativo-incremental cotidiano, na busca por melhoria contínua.

A realização da estratégia não pode ser tratada como um projeto *waterfall*, em que temos um plano de médio prazo e acompanhamento periódico previsto *x* realizado, auditado e reportado. É preciso ver, entender, adaptar e superar.

O livro tem foco nos líderes, mas minha percepção é de que no mundo ágil temos círculos concêntricos de execução, desde as pessoas, equipes, times, áreas, até o executivo. A execução é responsabilidade de cada um, posto que em agilidade todos somos protagonistas daquilo que construímos.

10.2 PDCL Ágil

"Frente à inovação, se nos mostram algo conhecido, ficamos sossegados.
– Friedrich Nietzsche

No ritmo que as coisas vão, em breve ninguém vai perguntar se você é ágil ou não, pois não haverá mais sentido nesta pergunta: todos seremos ágeis. Se você é de TI, você tem que ler e avaliar Scrum, XP, LSD, FDD, DDD, TDD, Crystal etc., mas se não é, está com sorte! Basta entender os princípios e o que é PDCL, um framework com mais de 50 anos, enxuto e ágil por natureza.

Com ciclo iterativo-incremental de Deming, o conhecido PDCA (*Plan-Do-Check-Act*), alterando o A por L (*Learn*), veremos que isto é suficiente para melhorar o trabalho de qualquer área (tesouraria, redação, distribuição, marketing, planejamento, departamento de pessoal, financeiro, multimídia), desde que selecionemos boas práticas que o suportem.

Para efeito visual e estabelecimento de um modelo mental, proponho a inclusão de duas informações adicionais ao ciclo. Entre o Learn e Plan vamos incluir um marco de Visão para esclarecer que é fundamental ter um alinhamento estratégico e definição de metas que vão além daquilo que estamos construindo naquele momento. Outra inserção, entre o *Do* e *Check*, deve ser uma Daily, pela importância de haver um alinhamento de execução no mínimo a cada 24 horas:

Fig. 10.1: PDCL Ágil

Visão

É fundamental se utilizar de boas práticas de refinamento de visão e estratégia, bem como da convergência de técnicas como Business Model Canvas, Stream Value Mapping, User Story Mapping para entender e perceber sonhos e superpoderes desejados para o negócio, produto ou serviço. Também é importante manter dinâmicas de *business brainstorming*, evitando atos isolados e favorecendo decisões coletivas e amplificadas. Estratégia não é algo que se faz sozinho:

- Admita que todos podem ter boas ideias;
- Senso de pertença e engajamento se conquista mais fácil com participação efetiva;
- Discurso e prática, transparência é o melhor caminho;
- Somente conhecendo e acreditando nos engajamos.

1º P (Plan)

Planejamento: em períodos recorrentes de no máximo 1 mês, é preciso parar e planejar o próximo período, discutir o que temos pela frente, equipe, atividades, datas-marcos, riscos e oportunidades. Essas discussões servem para agregar o que já aprendemos, além de oferecer sentimento de propriedade ao time, de posse das informações, cientes de seu papel e envolvimento:

- **Toda** equipe reunida para estimar o próximo ciclo;
- Sugiro o uso de um quadro branco ou folhas A1;
- Evite atividades pequenas ou grandes demais;
- Separar visualmente o que é projeto de operação;
- Defina metas e métricas para depois acompanhar;
- É importante esclarecer as prioridades e contingências.

2° D (Do)

Realização: a execução das atividades e tarefas planejadas e mesmo extras deve ser diariamente acompanhada de uma breve reflexão se estamos no caminho certo para cumprir datas e objetivos acordados. Todo o time deve diariamente buscar antecipar entregas ou corrigir desvios, fluindo estas informações aos demais interessados, para evitar surpresas:

- Focar na auto-organização para atingir os objetivos;
- Colaboração para garantir entregas frequentes;
- Multidisciplinaridade merece investimento;
- Não pode haver surpresas, quer para o bem ou mal;
- Dissemine conhecimento, motivação é tudo;
- Evite fazer mais do mesmo, tente fazer diferente.

Daily

Assim como no Scrum, é fundamental o uso de um quadro de tarefas para acompanhamento de evolução de status do trabalho, que será intensamente usado em reuniões diárias ou frequentes, posteriormente dissecado durante a retrospectiva, na permanente busca por pontos de melhoria, mitigação de riscos e aproveitamento de oportunidades.

3º C (Check)

Acompanhamento: a opção por um quadro de tarefas, com status e métricas ou o tipo que mais se adequar para a gestão visual conforme suas características, com o planejado ou não. É fundamental para que se exerça diariamente o modelo-mental de compartilhar os objetivos, realismo, transparência, inspeção e adaptação, eliminando estoque e surpresas.

- Use gestão visual e evite congestionar com supérfluos;
- É importante que transpareça a realidade de todos;
- Não espere o modelo ideal, comece e refatore;
- É importante o uso de selos especiais;
- Pode ter linhas por pessoa, projeto, prioridade etc.

4º L (Learn)

Retrospectivas: a cada fechamento de período, a equipe deve se reunir para aprender com o acúmulo de suas experiências. Isso é fundamental para a premissa de melhoria contínua de um processo iterativo-incremental como este. Refletir e permitir que o próprio time organize-se quanto a continuar, corrigir, começar ou abandonar práticas, de acordo com o que estas agregam.

10.3 Estratégia para inovação

"O trabalho da direção não é a gerência, é a liderança."
– William E. Deming

Volto ao conceito de *exploitation* e *exploration* relativos à teoria da organização baseada no aprendizado e no conhecimento. Trata-se do ideal de equilibrar recursos ao longo do tempo entre a aprendizagem gerada pela exploração de algo conhecido ou na busca da inovação competitiva.

O objetivo da busca pela inovação é criar valor para o negócio sob diferentes formas, em melhorias incrementais de produtos existentes, com o desenvolvimento de novos produtos e serviços, redução de custos, aumento da eficiência, novos modelos de negócios, novos empreendimentos, entre outras formas.

Busque inspiração no *Design Thinking*, no seu duplo diamante, conceito também utilizado na base do *gamestorming*. O método de criação da inovação é criar funis imaginários, idear, descobrir, selecionar e desenvolver, para então refiná-los em formas úteis, buscar ganhos, aumentar eficiência, reduzir custos. Estratégia de inovação é estabelecer boas práticas que alimentem e processem o funil.

Se você acha que o primeiro passo é gerar muitas ideias, talvez você esteja olhando para o lado errado. Assim como uma mina de minério de ferro, há muito trabalho prévio à retirada do primeiro quilo das profundezas da terra. Temos de começar o processo de inovação com o pensamento estratégico, para assegurar que ideias serão bem trabalhadas e que estarão alinhadas com a estratégia.

1º. Pensamento estratégico

O processo de inovação começa com o objetivo de criar um diferencial competitivo, sobre como a inovação vai agregar valor aos seus propósitos estratégicos, incentivando este exercício nas áreas onde a inovação tem o maior potencial para conversão desta energia em vantagem estratégica.

2º. Gestão de portfólio de inovação

Toda inovação está sujeita a dar certo ou errado. É fundamental administrar carteiras de inovação, equilibrar os riscos inerentes ao desconhecido com as recompensas alvo de sucesso, gerando hipóteses, validando, arriscando, aprendendo, falhando, tomando decisões.

3º. Pesquisa

Nas pesquisas, trabalharemos diferentes tecnologias, mudanças sociotécnicas, expondo as oportunidades para a inovação. O resultado de uma boa gestão de portfólio de inovação é uma carteira de inovação, uma mistura de projetos de curto e longo prazos nos quatro estágios do ciclo de inovação: idear, prototipar, descobrir e modelar.

4º. Insight

Este processo de inovação difere da geração fortuita de ideias. É o resultado de uma estratégia bem aplicada para a geração e desenvolvimento de proposições inovadoras, é o resultado de equipes e pessoas instigadas a fazê-lo de forma a perceber e preencher oportunidades;

5º. Materialização

Chega a hora da construção, posto que sou agilista, cabe relembrar o uso de boas práticas de MVP, ciclos iterativos-incrementais, equipes multidisciplinares, foco em valor, eliminação de desperdícios e nos princípios ágeis aplicados a UX, desenvolvimento, qualidade, resultados.

6º. Desenvolvimento do cliente

Momento de atender uma demanda órfã ou trabalhar para criar uma nova demanda, por uma nova solução ou produto inovadores. É a apresentação e compreensão desta inovação em busca de um crescimento rápido e sustentável das vendas.

7º. Aceitação

É chegada a hora do *Exploitation*. Agora que ganhamos o retorno financeiro com a venda de sucesso dos novos produtos e serviços, vamos colher os benefícios e trabalhar para gerar ciclos de aumento da eficiência e produtividade, persistindo e ampliando pelo máximo de tempo o investimento.

A tempo, quer a seleção, pesquisa e desenvolvimento tenham gerado bons resultados ou não, está na hora de iniciar um novo ciclo de *Exploration*.

10.4 Manifesto Ágil ajustado para outras áreas

"Quem nunca errou nunca experimentou nada novo."
– Albert Einstein

Para treinamentos de equipes de áreas de negócio, backoffice, fornecedores, parceiros, clientes, apresentar o Manifesto Ágil original é antecipadamente gerar uma barreira, pois ao ler as palavras software e desenvolvimento, as pessoas blindam sua forma de trabalhar como "diferente".

A mesma coisa ocorre se usarmos o nome Scrum ou seu ciclo de vida, com papéis, as timeboxes, artefatos e regras, pois mesmo que na hora não se rebelem, os mais interessados irão para a internet procurar mais informações e receberão uma avalanche de artigos e exemplos de TI/Software.

O que estou querendo dizer é que o Manifesto Ágil também serve para backoffice e áreas de negócio. Em treinamento de equipes de outras áreas, eu tomo a liberdade de fazer pequenos ajustes: indivíduos e interação entre eles mais que processos e ferramentas; partes do trabalho concluído mais que documentação abrangente; colaboração com o cliente mais que negociação de contratos; responder a mudanças mais que seguir um plano. Você pode rever os princípios ágeis em 3.4.

10.5 Um PDCL no financeiro

"Para mover o mundo, o primeiro passo é mover a si mesmo."
– Platão

Realizei treinamentos e workshops com diversas áreas de negócio e corporativo, redação, *database marketing*, financeiro, planejamento, inicialmente para alinhamento, mas passei a assistir equipes de backoffice e de negócio adotando o ciclo PDCL Ágil e partindo para a execução.

Precisamos entender o cotidiano e desenhar um quadro de tarefas adaptado. Neste exemplo do financeiro estamos na terceira semana do mês, as 2 últimas ainda aguardam no planejado, a 3ª semana (em curso) está no quadro central e as últimas 2 primeiras já estão no de Realizado do mês:

- Quadro de planejamento mensal: o quadro possui 5 linhas (semanas) por 7 colunas (dias da semana) – A área financeira tem um pack de atividades recorrentes (fechamentos, conciliações, pagamentos, conferências etc.). No início de cada mês, ao fazer a análise de feriados e dias úteis, redistribuem-se os post-its correspon-dentes nas datas em devem ser executadas e limites.
- Scrum board: no início de cada semana do mês, traz-se toda a linha da semana que inicia do quadro de planejamento mensal para a coluna *TO DO* do quadro de tarefas.
- Quadros auxiliares: Teremos pequenos quadros auxiliares, para impedimentos, lembretes para retrospectiva, pacto de time, ausências (férias e feriadões programados) e o que mais o time desejar tornar visível.
- Quadro com o Executado Mensal: Igual ao quadro de planejamento, mas este, a partir do final da primeira semana, receberá as atividades concluídas no Kanban, na coluna *DOING*, que devem ser movidas para a linha da semana que acabou no quadro de execução mensal.

Faz-se uso de post-its contendo data-marco, tempo estimado de dedicação para realizar a atividade, data limite para início e selos para identificar atraso na execução, extras e impedimentos, podendo inclusive ter-se no verso registro dos dados básicos mês a mês. Dessa forma, manteremos todas as informações, o que proporcionará aprendizado e oportunidades para melhoria contínua.

10.6 Rainforest

"Delegar também é uma forma de educação."
– Kaoru Ishikawa

A construção de organizações baseadas em profissionais e times com grande senso de pertença em um substrato de liberdade com responsabilidade atrai e retém talentos. Nessas condições, mudar não exige uma revolução, basta um aproveitamento que transforme GigaWatts de energia estática e latente em energia corrente.

Hwang define parques tecnológicos como uma *RainForest*:, pela capacidade de gerar inovação a partir da interação entre os mais diferentes organismos que a compõe. É aumentar a atração e retenção de talentos através de ambientes abertos e valores coletivos baseados em desafios, crescimento e sustentabilidade.

Existe um mínimo de confiança, orgulho, parceria naquilo que você faz todos os dias? Você se sente motivado a se envolver com novos conhecimentos, contribuindo naquilo que você se acha capaz de fazer? Você faz mais do mesmo há quanto tempo ou tem espaço para ser proativo e cooperativo em fazer melhor?

Qual a sua relação com seu chefe, você consegue dizer o que pensa? O quanto você pode discordar e debater sobre ações que você acredita não serem o melhor para a empresa, para sua área, para seu projeto e sua carreira? O quanto tenta mudar, propondo, argumentando, o quanto faz acontecer?

Se você olhar para os lados, vê brilho nos olhos da galera? E nos seus? Se você olhar para você mesmo há três anos, o quanto você cresceu, melhorou, conseguiu mostrar que é capaz e merece ser valorizado? Os momentos de descontração (se eles existem) são para pensar em melhorias e futuro ou são para reclamar do passado e presente?

Capítulo 11

Gestão do conhecimento

Por mais que a adoção seja um momento importante de mudança de modelo mental para começar a rodar um novo processo, é o continuum iniciado a partir dela que significará sucesso ou não. Relembrando a curva de Tuckman, a adoção exige esforço mas é breve, o dia seguinte é que são elas, pois exigirá de fato internalização dos princípios e conceitos. A prática diária é o maior desafio para a conquista da melhoria contínua.

11.1 Gestão do conhecimento

"Tudo deveria se tornar o mais simples possível, mas não simplificado."
– Albert Einstein

Takeuchi & Nonaka, os precursores do Scrum também são os pais da Gestão do Conhecimento (1995), disciplina que envolve cultura e valores: *"Em-*

bora nós usemos o termo [criação] de conhecimento organizacional, a organização não pode criar conhecimento por si própria sem a iniciativa do indivíduo e a iteração que acontece dentro do grupo."

Os professores Ikujiro Nonaka e Hirotaka Takeuchi conceberam no livro *The Knowledge-Creating Company* a espiral do conhecimento, também conhecida como modelo SECI:

- Socialização – Tácito/Tácito – Compartilhamento de conhecimento entre as pessoas, cara a cara, interno ou externo, como assistir palestras, bate-papos, observação, conferências, eventos, replicação, brainstorming; trata-se da transmissão no dia a dia;

- Externalização – Tácito/Explícito – Conceituar o conhecimento passado tácito para explícito, através de registro e documentação, descrevendo um processo, técnicas, desenhar um diagrama, metáfora e analogias. É a fase mais importante;

- Combinação – Explícito/Explícito – Sistematizar o conhecimento e distribuí-lo para gerar educação organizacional, trata-se da união de diferentes conhecimentos explícitos já externalizados gerando um novo conhecimento;

- Internalização – Explícito/Tácito – Transformação de conhecimento explícito em conhecimento tácito para a geração de inovação, novos modelos mentais, compartilhado, iniciando nova espiral de criação de conhecimento, interagindo, formando opinião e aplicando o novo conhecimento.

Fig. 11.1: Modelo SECI

Gestão do conhecimento não é uma tarefa, mas um traço cultural da empresa que deve ser tratado e conduzido de forma que todos entendam a importância de estabelecer o registro mínimo necessário, contra a realidade do desperdício a cada troca de integrante, novo projeto ou tecnologia.

11.2 Conceito de *Ba*

"Inovação vem de pessoas que se divertem com seus trabalhos."
– William E. Deming

Além do modelo SECI, Nonaka e Takeuchi desenvolveram o conceito de *Ba*, espaços para a geração de conhecimento. Chamamos de *Ba* cada espaço momentaneamente compartilhado para geração de conhecimento, de forma consciente e organizada, mesmo uma conversa na hora do intervalo, um café. Se investido de contexto visando o debate, a troca, um *Ba* pode ser físico, virtual ou mental:

- Um espaço *Físico* como escritório, espaço de negócios;
- Um espaço *Virtual* como e-mail, teleconferência etc.;

- Um espaço *Mental* como ideias, valores, objetivos, experiências.

O *Ba* não é um valor objetivo, mas subjetivo e dependente dos atores que o constituem. Cabe à organização proporcionar as condições, incorporar estes valores em seu modelo mental e de seus integrantes:

Originating *Ba* (tácito - tácito)

São espaços que originam o conhecimento através da interação pessoal de seus participantes, pela soma de percepções, vivências, aprendizado, sentimentos etc. Reconhecemos ali a socialização do conhecimento tácito individual em coletivo, pelo compartilhamento de conhecimento, habilidade e atitudes. Aqui teremos a gênese do sentimento de grupo, cada um colaborando de forma particular.

Dialoguing *Ba* (tácito - explícito)

São espaços que oportunizam a conscientização e externalização do conhecimento pertencente a cada indivíduo na construção de conceitos ou contextos comuns, como para a definição de produtos e serviços, processos e construções de interesse comum. Aqui há um processo der conversão do conhecimento tácito em conhecimento explícito, do conceitual para a criação de ativos de conhecimento.

Systemizing *Ba* (explícito - explícito)

Espaços físicos ou virtuais com uma natureza coletiva interessada em gerar conhecimento a partir da combinação daqueles já explicitados entre seus participantes. É importante para construir novos conhecimentos para a organização, partindo do geral para o específico, utilizando-se de comunidades de conhecimento, redes, compartilhando o resultado desta construção na forma de documentação, licenças, repositórios.

Exercising Ba (explícito - tácito)

Espaços que compartilham e que retroalimentam o conhecimento gerado e formalizado pelo grupo ou organização novamente para os indivíduos, pos-

sibilitando a aplicação deste conhecimento geral em novas conversões de melhorias aos processos e conceitos vigentes. Contribui para aprimorar o trabalho e serviços, gerar ou refinar know-how, evoluindo processos produtivos, administrativos ou de serviços.

11.3 Comunidades de prática

"Faça o elemento humano tão importante quanto os elementos técnicos e de negócios."
– George Kembel (Stanford)

Gestão do conhecimento é uma disciplina complementar às metodologias ágeis que oferece artefatos e técnicas para a construção e disseminação de aprendizados e conhecimento. A criação de grupos e reuniões, intraorganizacionais e interorganizacionais, garante a propagação de teorias e boas práticas, sucessos e fracassos, contribuindo na construção de um vórtice contínuo de crescimento.

Dez equipes ágeis que não se falam e não compartilham suas experiências, não gerarão sinergia e aprendizado vicário. As tentativas, erros e aprendizados de uma podem e devem otimizar a curva de aprendizado de outras. O instanciamento do modelo SECI elimina desperdícios e gera valor.

São grupos que interagem periodicamente com foco em uma área de interesse, campo de conhecimento ou profissão específica. O termo *Comunidades de prática* foi cunhado por Jean Lave e Etienne Wenger no início da década de 90. Uma CoP pode ser no mundo real ou virtual, mas quero destacar comunidades presenciais em empresas, quando compartilham, debatem, experimentam e geram conhecimento.

Incentivar a criação de comunidades de prática dentro de sua empresa e facilitar que seu pessoal participe de grupos de usuários e comunidades de práticas é uma das formas mais eficazes de gerar espirais de compartilhamento e gestão de conhecimento, otimizando o aprendizado vicário.

Aprendizado vicário é uma maneira de acelerar nosso aprendizado a partir da observação percepção dos erros e acertos dos outros, é perceber riscos e encontrar alternativas e soluções a partir da interação e da colaboração

com outras pessoas que vivem o mesmo contexto ou enfrentam o mesmo problema.

Interorganizacional

Se me perguntarem os porquês da participação em grupos de usuários e comunidades de prática abertas, fácil: reúnem-se ali pessoas de iguais interesses, inquietas, com fome de conhecimento, interessadas em interagir com pessoas com variadas expertises e vivências, dispostas a compartilhar tanto inquietações quanto convicções.

As pessoas que colaboram em GUs e CoPs, com frequência, são mais valorizadas, pois agregam outros prismas, argumentos e contra-argumentos, saem do seu quadrado, ousam e se aventuram. Se conseguem alçada, promovem reuniões equivalentes dentro da sua empresa, levando suas boas práticas de Gestão do Conhecimento.

Intraorganizacional

Vamos considerar uma equipe tradicional de testes de software, chamada por algumas empresas de célula de testes ou fábrica de testes, mas que se vê em meio a um processo de adoção Scrum. Na composição de equipes ágeis, podemos ter a constituição de equipes multidisciplinares, colaborativas, com Product Owner, analista, desenvolvedor e testador.

Em muitos casos, o ganho da aproximação física da equipe de projeto envolve também um afastamento de seus pares, mas afinal: o quanto é possível aproveitar o melhor de cada um destes dois modelos? Isto é, aproximação e integração no time ágil ao mesmo tempo em que reforçamos vínculos entre iguais, seus pares.

A criação de comunidades de prática envolve compartilhamento de repertório, auto-organização e objetivos convergentes. A participação tende a oferecer uma visão maior do seu projeto e processo, pois gera debate em um plano evolutivo que vai além do seu dia a dia e contexto atual. O senso de pertença e melhoria Kaizen oferecido às equipes Scrum é complementar aos resultados de uma CoP.

Acrescente a isso alguns ganhos como integrar rapidamente novos colaboradores, diminuir curvas de aprendizado, conhecimento e alinhamento as

boas práticas, autoconhecimento, autodiagnostico, repasse de know-how, alinhamento de riscos e oportunidades, favorecendo o exercício da discussão de ideias, negociação e outras competências.

Ilhas e arquipélagos

Cuidado, portanto: adotar Agile e não instanciar a Gestão do Conhecimento proposta pelo Modelo SECI e pelas Comunidades de Prática é um risco de gerar ilhas, um arquipélago em que todos os times correrão o risco de cometer os mesmos erros. O esperado é usar a GC para que o aprendizado dos times possa potencializar ao máximo o aprendizado organizacional. Pense nisso!

11.4 Agile subway maps e dashboards

"Você não tem que escolher entre fazer o que você ama e ganhar a vida."
– George Kembel (Stanford)

Qualquer viajante no mundo sabe da importância de termos um bom mapa de metrô na mão quando visitamos uma metrópole, quer em Buenos Ayres ou Paris. A partir das estações e linhas do metrô, é possível planejar com facilidade e sucesso todos os movimentos. Acontece o mesmo no trabalho, quando sabemos quais as tecnologias, métodos e técnicas que temos à nossa disposição, qual o horizonte e nossos sonhos.

Eu sempre usei diagramas modificados do Scrum para este fim, como um mapa de necessidades, oportunidades e sonhos, explicitando técnicas desejadas em cada momento, papel, timebox, artefato e regras. A proposta é simples: reproduza um diagrama Scrum ampliado em escala maior possível, estabeleça o debate sobre o uso iniciante, pleno ou ninja e estabeleça um *canvas* ou *dashboard* para registro de planos de ação e evolução de cada equipe participante.

Tenho falado muito sobre comunidades de prática (CoP) como instrumento de gestão de conhecimento em meio a processos de aculturação ou de melhoria contínua, como por exemplo um mapa de melhores práticas e técnicas a ser utilizado por uma CoP de Scrum Masters, PO, QAs ou desen-

volvedores de forma a visualizar-se o status atual de cada equipe, bem como a disseminação de conhecimento e próximos passos.

Até aqui sempre funcionou muito bem, mas recentemente ouvi falar destes diagramas com a galera do SERPRO e não me aquietei até buscar fontes e artigos relacionados. Parafraseando Lavoisier, nada se cria e nada se perde na TI, apenas transformamos a fim de acoplá-las a nossas organizações. Que tal um conceito de mapa mental na forma de um mapa de estações de metrô?

Com cada time e integrante se inspira neles para analisar seu status e evolução em uma perspectiva Kaizen, auto-organizada. Seria fácil entre pares ali pactuar e visualizar o que é básico, intermediário e avançado, o que é mínimo, necessário e desejável.

Imaginem uma CoP de analistas de testes, QAs e testers mapeando o que é *commodity*, o que é meta e o que é sonho, permitindo um autodiagnóstico e a identificação de gargalos e experimentações que estão dando mais ou menos certo entre as diferentes equipes da organização.

É instanciar o modelo SECI de Takeuchi e Nonaka, aproveitando para trazer o conceito de *exploitation* (dominado) e *exploration* (inovação), ambos essenciais para entender e instanciar o conceito Kaizen de melhoria contínua.

Ninguém é uma ilha, equipes ágeis menos ainda. Equipes-ilhas jamais serão ágeis, porque não entenderam que interação limitada a seus colegas de equipe não lhes dará oportunidades de aprendizado vicário e compartilhamento, de orgulho de fazer parte de algo muito maior que o seu Kanban.

11.5 Tipos de eventos

"Qualquer tecnologia suficientemente avançada é indistinta de magia."
– Arthur C. Clarke

É bom irmos nos acostumando com as nomenclaturas e oportunidades de eventos básicos e dinâmicas para geração ou passagem de conhecimento, sabendo escolher o formato certo para potencializarmos o resultado. A ideia não é detalhar, nem listar todos, mas provocar quem lê a acreditar em si mesmo e fazer acontecer. O formato e o tempo é você que vai definir conforme sua disponibilidade e necessidade:

- **Café mundial** – Usado em locais públicos onde organizamos mesas conforme interesses em temas selecionados, com quórum máximo onde as pessoas se candidatam ou são sorteadas, com rodízio controlado.
- **Debate** (60 minutos) – Caracteriza-se pela discussão entre duas ou mais pessoas, com diferentes pontos de vista, com moderador ou mediador; a plateia pode participar por escrito com mediação ou direto.
- **Coding Dojo** (120 minutos) – Uma sofisticada técnica colaborativa de aprendizado, onde um desafio lógico é apresentado e um grupo de 10 a 15 pessoas tentam solucioná-lo, com um só computador usado em rodízio e dinâmica preestabelecida.
- **FishBowl** (60 minutos) – Debate participativo em grandes grupos, até 60 pessoas; no centro, um espaço com 5 cadeiras, iniciando com 4 ocupadas e 1 vazia. Durante o debate, sempre que alguém da plateia ocupar a vazia, um dos outros 4 volta a plateia.
- **Workshops** (50 minutos) – Realização de dinâmicas sobre práticas, princípios e valores, buscando reproduzir em um ambiente controlado as mesmas situações vividas no trabalho, proporcionando aprendizado lúdico com nossos acertos e erros.
- **Lightning Talks** (15 a 20 minutos cada) – Palestras rápidas de 10 ou 15 minutos, pré-agendadas ou abertas para pessoas interessadas se candidatarem e falarem sobre assunto de seu domínio e interesse. Ao final, há 5 minutos para perguntas.
- **Mesa Redonda** (120 minutos) – De 4 a 8 especialistas de referência, com regras claras e pré-acordadas, moderador e um assunto polêmico, intercalando 10 minutos de introdução e 2 para cada questões e respostas, com direito a réplica e tréplica.
- **Open Space** (90 minutos) – Debates em grandes grupos, que propõem e votam nos temas, para constitui-ção de subgrupos que debaterão e apresentarão um plano de ação. Conclui-se com votação do melhor ou compilação de um por todos.

- **Painel** (90 minutos) – Caracterizado por um orador, um moderador e até 4 painelistas de reconhecido saber sobre o tema central mas com opiniões parcial ou totalmente divergentes, que debaterão sua ideias.
- **Palestra** (30 a 60 minutos) – Apresentação de certo tema a um grupo que possui algum conhecimento prévio sobre o assunto, com um coordenador para triar perguntas encaminhadas por escrito ou feitas diretamente ao palestrante.

11.6 Hackatona PDCL

"Se você deseja um trabalho bem feito, escolha um homem ocupado; os outros não têm tempo."
– Benjamin Franklin

Imagine um dia dedicado a uma sequência lúdica de Agile Games baseados em um projeto real ou hipotético, 20 pessoas separadas em três equipes com variados papéis em imersão com visão, mapping, planning, daily, review e retrospectiva.

É uma oportunidade de se aprender fazendo, uma imersão em um projeto, usando canvas, mocando, simulando discovery e delivery. Com um mínimo de teoria, o suficiente para rodar, com muita interação e momentos para debate e melhoria.

O fundo de cena é um projeto real, iniciado com um quebra-gelo, um *briefing* sobre o contexto, características, necessidades, clientes, objetivos, restrições e tudo o mais.

1) O primeiro passo é entender o projeto, uma dinâmica de Visão, objetivos e clientes, trabalhando bem a importância e entendimento da estratégia;

2) O próximo passo é uma User Story Mapping, um debate entre todos os integrantes na montagem de um mapa das User stories, sprints e releases;

3) A seguir vem uma construção simbólica, iterativo-incremental, com mockagem em papel, usando as paredes como mundo real e um Kanban demonstrando o fluxo de status de cada tarefa;

4) Seguindo o clima de games, jornadas, intercalando momentos para feedbacks. Cada etapa exige equilíbrio entre planejamento, execução, feedback;

5) Review, último baluarte dos intermediários da verdade entre os usuários e equipe. Alinhamento do road map, riscos, oportunidades, valor, velocidade e qualidade;

6) Retrospectiva, feedback e autoconhecimento para melhoria contínua; entender o que deu certo ou nem tão certo assim, planos de ação.

11.7 Colaboração é a menor distância

"A maior responsabilidade de qualquer líder é capacitar os outros a fazer as coisas."
– Tom Kelley (IDEO)

"Ágora", na Grécia Antiga, era um espaço aberto onde todos os cidadãos podiam expor suas ideias, sugestões e propostas; um espaço para inclusão, de ciências, artes e cultura cidadã e colaborativa. Colaboração é a menor distância para a validação da maioria de nossos pressupostos. Desde 2011 venho insistindo para os colegas criarem blogs, participarem de eventos, apoiarem G's (grupos de usuários), contribuírem em comunidades de conhecimento, quer seja através de artigos, comentários ou audiência; todos os papéis são importantes.

Ao perceber algo construtivo, participe, instigue-se a ser a próxima onda. Há muitos formatos de eventos. Sempre aprendemos algo novo nessas ocasiões e o Google se encarrega de mostrar outros tantos; é só querer.

A seguir, compartilho 10 dicas de como preparar e compartilhar com sucesso seu papel como apresentador, palestrante, instrutor ou debatedor e, na sequência, apresento 10 formatos de eventos para você organizar na sua empresa ou participar:

Uma pesquisa prévia amplia resultados

Procure antecipar informações sobre o quórum, local, instalações, perfil dos participantes, outros palestrantes, faixa etária, equipamento, conexão de

internet, expectativas. Antecipe-se, ajuste o conteúdo a situação; cada caso é um caso e uma mensagem personalizada é mais bem captada e entendida.

Estamos em 2015, use a nuvem e prezi.com

Eu uso Prezi na nuvem, freemium, que lhe oportuniza palestras remotas, edição colaborativa, navegação e transição em "voo pássaro", com imagens, áudios, vídeos, links como se estivessem espalhados em uma gigantesca folha em branco e layers para mergulhar, ampliar, refinar.

Material diferente para tempos diferentes

Não use o mesmo material para diferentes tempos. Não deixe nas telas mais do que o estritamente necessário, a apresentação é para ser uma referência visual. Se houver textos, as pessoas não estarão prestando atenção em você, mas vão se frustrar tentando ler o amontoado de dados que você colocou lá.

Use as três memórias

Tem pessoas que percebem, entendem e memorizam melhor pela comunicação visual, através de imagens e vídeos. Há outras que se utilizam melhor da audição, através de uma voz confiante, interessante, instigante, música. E há os que apreendem melhor pela ação, através de dinâmicas, games e mãos na massa.

Chegue cedo, prepare tudo e relaxe

Nunca deixe nas mãos de Murphy, evite chegar "na hora". Chegue cedo, instale, confira tudo, tenha mais de uma cópia, use um runtime local, nem sempre há internet. Dê o máximo mesmo que seja gratuito, pois a partir do momento que você aceitou ou organizou, é seu nome e de sua empresa.

Vícios de linguagem e outros mais

Conheça seus vícios de linguagem e treine, peça opiniões, filme suas apresentações, ganhe consciência e faça um esforço para controlar os *né*, *é*, *uhhhh*,

olhares, movimentos, mãos no bolso ou coçando algo. Seu corpo fala mais que sua boca e, para corrigir, precisamos ter consciência e querer melhorar.

Eles foram lá para aprender algo

Evite ficar 15 minutos falando de seu currículo, afinal não é uma entrevista de emprego, a galera foi lá para aprender algo sobre o tema divulgado. Seja divertido, mas não faça um espetáculo, alguns se perdem em meio a piadas e causos pessoais. Seja interessante e instigante no tema, não saia dele.

Se algo der errado, siga em frente

Seja sincero, mas acima de tudo seja construtivo, evite se perder em explicações ou desculpas. Se algo der errado, siga em frente. O importante é ter contingências e estar preparado para o improviso.

Registre tudo

Registre todo o processo desde o início, poste, divulgue, tire fotos, filme, tenha um tripé, melhor ainda, um companheiro para cada viagem, porque sozinho ninguém faz nada e seria desperdício não publicar informações adicionais, links, textos, depoimentos no blog, site, newsletter, facebook.

Seja feliz!

Falar em público ou escrever é um grande barato, relaxe, faça o seu melhor, convide os amigos, sempre tire fotos, compartilhe nas redes, valorize assuntos sobre coisas em que você acredita, pois passar crenças e princípios é melhor que passar conteúdo. As pessoas percebem se você está no automático.

11.8 Eventos, confrarias e voluntariado

"A estratégia é uma força poderosa na determinação dos resultados."
– Michael Porter

Os maiores exemplos são os grupos de usuários da SUCESU, comunidades de prática como o TecnoTalks no TecnoPUC, eventos internos ou abertos

de empresas que entendem que conhecimento é um ativo vivo e que ao tentar isolá-lo ele caduca; é preciso fomentar a troca, geração e retroalimentação.

Há uma infinidade de oportunidades de colaboração, iniciativas como o RHoK (*Random Hacks of Kindness*) ou junto a comunidades carentes no entorno das empresas, seus bairros e vilas, quer desenvolvendo sites e serviços de forma voluntária ou arrecadando leite, alimento ou agasalhos e distribuindo em creches, associações ou postos.

Oportunidades estas que, sem isso, ficam restritas a 5% de participantes de instituições seculares como o escotismo, Lions, Rotary, algumas ONGs e que, com um pouco de iniciativa, torna-se possível a cada evento ou período convergir contribuições que quase nada custam e que podem fazer a diferença em uma consciência coletiva mais cidadã.

O voluntariado pressupõe pleno exercício da cidadania, quando de forma livre e organizada realizamos algo em prol do próximo para resolução de carências ou problemas. Pode ser realizado por iniciativa individual, coletiva ou institucional, a favor de alguém, grupo ou comunidade, conhecido ou desconhecido, devido a infortúnio ou carência.

A ONU convocou o mundo em oito objetivos do milênio:

- Acabar com a fome e a miséria;
- Educação básica de qualidade para todos;
- Igualdade entre sexos e valorização da mulher;
- Reduzir a mortalidade infantil;
- Melhorar a saúde das gestantes;
- Combater a AIDS, a malária e outras doenças;
- Qualidade de vida e respeito ao meio ambiente;
- Todo mundo trabalhando pelo desenvolvimento.

11.9 Acho que aprendi algo novo, e agora?

"O verdadeiro brilho do processo de design centrado no ser humano é que ele nos mantém humildes."
– Susie Wise (Stanford)

A pergunta a ser feita ao final de cada livro, evento, palestra, debate, workshop, é: *"O que vai fazer com o que aprendeu?"* Se a resposta for vaga, só um elogio ou algo no gerúndio, pensativo, sinto muito, mas então foi apenas estoque, desperdício. Devemos arregaçar as mangas e validar o que foi visto, envolver outras pessoas, questionar, aplicar.

Por outro lado, devemos equilibrar todo tipo de eventos, de forma que tenhamos tempo para absorver e praticar. Mas cuidado para não tornar algo bom em algo compulsivo e estressante, entre um evento e outro, administre os conteúdos e dê-se tempo para consolidar o que viu em três passos básicos:

1) **Comunicação** – Um primeiro passo sustentável é replicar aos colegas. Não precisamos ser professores para montar um resumo, palestra ou lightning talk sobre o(s) tema(s) visto(s) em um evento ou workshop;

2) **Fixação** – Se colocarmos de alguma forma em prática aquilo que aprendemos, reteremos muito mais, validaremos mais, e mais ainda se o fizermos coletivamente, entre pessoas que têm os mesmos objetivos. Juntos aprofundamos os temas sob diferentes abordagens;

3) **Replicação** – O resultado prático é maior se revisitarmos e vivenciarmos através de grupos de estudos, debates, dojos etc., tanto no conceito japonês "Kata" ou na de Aristóteles, *"atingimos a excelência através da repetição"*. Apenas ter o insight não leva a nada.